EPSG Draft Standard 302-C

Ethernet POWERLINK

Part C: PollResponse Chaining

Version 1.1.1

B&R Industrial Automation GmbH

POWERLINK-Office
B&R Straße 1
5142 Eggelsberg
Austria

powerlink.office@br-automation.com
www.br-automation.com/en/technologies/powerlink/

Pre. 1 Disclaimer

Use of this EPSG Standard is wholly voluntary. The EPSG disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other EPSG Standard document.

The EPSG does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. EPSG Standards documents are supplied “AS IS”.

The existence of an EPSG Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the EPSG Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Users are cautioned to check to determine that they have the latest edition of any EPSG Standard.

In publishing and making this document available, the EPSG is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the EPSG undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other EPSG Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of the EPSG, the group will initiate action to prepare appropriate responses. Since EPSG Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, the EPSG and its members are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of EPSG Standards are welcome from any interested party, regardless of membership affiliation with the EPSG. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be sent to the address given on the page before.

Pre. 1.1 Patent notice

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. B&R shall not be responsible for identifying patents for which a license may be required by an EPSG standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Pre. 2 History

Vers.	Date	Author / Filename		short description
0.0.1	2009-05-29	Stephan Kirchmayer	B&R	created
0.0.2	2009-12-09	Stephan Kirchmayer	B&R	References to [2] added
0.0.3	2010-08-04	Stephan Kirchmayer	B&R	Feedback from TWP meeting in 3/10
0.0.4	2010-12-21	Stephan Kirchmayer	B&R	Change of category and description of RelativeLatencyDiff, D_DLL_CNPResChaining_BOOL and D_DLL_MNPResChaining_BOOL change to mandatory
0.0.5	2011-01-31	Stephan Kirchmayer	B&R	Add recommendation to use DestMacAddress Change state of document to DSP
0.0.6	2011-05-25	Stephan Kirchmayer	B&R	(Editorial) feedback from J. Poppe
1.0.0	2012-02-23	Stephan Kirchmayer	B&R	Change state of document to DS
1.0.1	2015-12-21	Stephan Kirchmayer	B&R	Changes from TWG meeting 12/15 Status change to DSP
1.1.0	2016-03-14	Stephan Kirchmayer	B&R	Status change to DS
1.1.1	2023-05-31	Stephan Kirchmayer	B&R	© B&R due to dissolution of EPSG

Pre. 3 Content

Pre. 4 Tables

Pre. 5 Figures

Pre. 6 Definitions and Abbreviations

Pre. 6.1 Definitions

reserved	Reserved bits shall be set 0 by the sender. The receiver shall not interpret such bits. It is not allowed to use reserved bits. Their use is reserved for further development or by extensions of this specification.

Pre. 6.2 Abbreviations

ASnd	Asynchronous Send (POWERLINK frame type)
CN	Controlled Node
CRC	Cyclic Redundancy Check
ID	Identifier
MAC	Media Access Control
MN	Managing Node
PReq	Poll Request (POWERLINK frame type)
PRes	Poll Response (POWERLINK frame type)
PResMN	PRes frame of the Managing Node
SoA	Start of Asynchronous (POWERLINK frame type)
SoC	Start of Cyclic (POWERLINK frame type)

Pre. 7 References

[1] EPSG Draft Standard 301 (EPSG DS 301), Ethernet POWERLINK, Communication Profile Specification

[2] EPSG Draft Standard 302-A (EPSG 302-A), Ethernet POWERLINK, High Availability Extension

[3] EPSG Draft Standard 302-D (EPSG 302-D), Ethernet POWERLINK, Multiple PReq/PRes

1 Introduction

The „PollResponse Chaining“ is a standard feature to increase the performance of POWERLINK networks.

Instead of PReq/PRes frames only PRes frames are sent. The data, usually sent by the Managing Node in the PReq frames, are mapped into the PRes frame of the Managing Node (PResMN).

However it is still possible to use conventional PReq/PRes nodes in combination with PResChaining nodes.

1.1 Advantages

Shorter cycle times in contrast to a conventional network for the same configuration.

Increased performance, especially if the structure of the network is a line topology.

Faster reaction time, because all output data are published within one frame (PResMN). If all Controlled Nodes synchronize their outputs to the PResMN, the reaction time is reduced by one cycle.

2 Principles

2.1 The POWERLINK cycle

2.1.1 PReq/PRes

The sequence of frames of a conventional POWERLINK cycle looks like in the following figure.

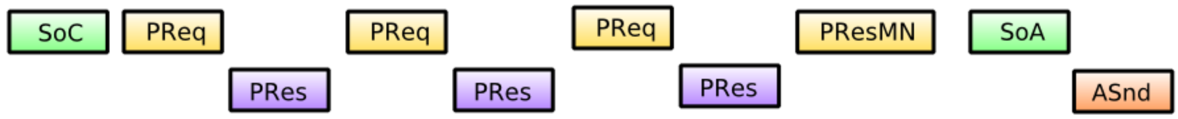

Fig. 1. POWERLINK cycle PReq/Pres

2.1.2 PRes Chaining

PRes Chaining does not use PReq frames anymore. The POWERLINK cycle time is reduced.

However it is still possible to use conventional PReq/PRes nodes in combination with PResChaining nodes.

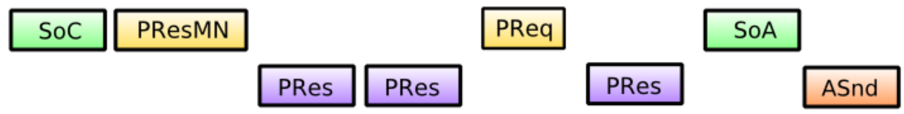

Fig. 2. POWERLINK cycle PResChaining

2.2 Time triggered sending of frames

In the conventional POWERLINK cycle a Controlled Node is only allowed to send a PRes frame after receiving its PReq frame. With PRes Chaining this rule is obsolete. Now the PRes frame is sent time triggered.

Each Controlled Node is configured by the Managing Node to send its PRes frame at a specific point in time (PRes Response Time, t_{PRes}). The time shall be calculated by the Managing Node in a way that the Controlled Node is still receiving the PRes frame of the predecessor at start time. The carrier sense mechanism of Ethernet delays the sending of the own frame until the current frame is completely received. By this PRes frames are sent with interframe gap delay.

The reference for the PRes Response Time is the end of the PResMN frame (after the CRC checksum).

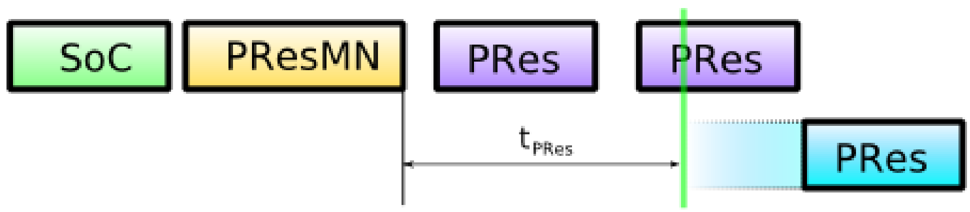

Fig. 3. Time triggered sending of frames

2.3 Features of performance

The overall size of output data by the Managing Node ist limited by the maximum data size of a PRes frame (=1490 bytes). However this limit can be exceeded if multiple PRes frames are used. See [3] for more details on Multiple PReq/PRes.

PResChaining nodes and conventional PReq/PRes nodes may be mixed in a POWERLINK cycle. Within a cycle PResChaining nodes shall always send their PRes frames before the Managing Node starts the polling of conventional PReq/PRes nodes.

PResChaining nodes and multiplexed PReq/PRes nodes may be mixed in a POWERLINK cycle. But multiplexing is not defined for PResChaining nodes.

PRes Chaining nodes always require space for their output data in the PResMN wether if such a node is present or not.

The Managing Node is not able to slow down the network anymore. It must be able to process the receiving frames in time.

3 Measuring the propagation time

3.1 General

To be able to configure the time triggered system, the propagation delays of the network have to be measured. This is done by the Managing Node sending SyncReq frames to each node. The addressed node replies with SyncRes frames immediately or within a constant time.

The SyncReq and SyncRes frames are sent with a dedicated multicast MAC address (SoA resp. ASnd) and so received from all nodes. Every node calculates the difference between the receive-times of SyncReq and SyncRes. Subtracting the length of the SyncReq from the difference calculated above results in the propagation delay between these two nodes based on the receive time of the SyncReq.

After polling all Controlled Nodes in the network with SyncReq frames, every node knows the relative propagation delay to every other node.

The result can be shown in a matrix, e.g. for a line topology:

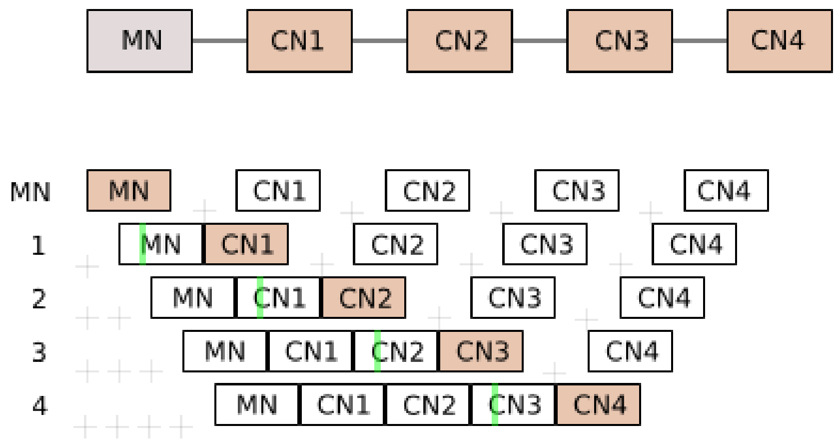

Fig. 4. Line topology with timing graph

If the propagation delay between two nodes is set to one the relative propagation time between the nodes results in the following table.

	MN	CN1	CN2	CN3	CN4
MN	x	2	4	6	8
CN1	-	x	2	4	6
CN2	-	**0**	x	2	4
CN3	-	0	**0**	x	2
CN4	-	0	0	**0**	x

Tab. 1 Propagation matrix for line topology

One line of the table indicates the times measured at this node. The times are the propagation delay, i.e. the difference between between the send- resp. receive-time of the SyncReq and the receive-time of the SyncRes minus the length of the SyncReq minus the response latency of the Controlled Node.

Only the relative propagation delays must be considered for the calculation of the PRes Response Time. (bold values in the matrix above)

The Managing Node shall calculate the PRes Response Time for each node considering

- all relative propagation delays of the predecessors
- the duration of all PRes sent before – except the direct predecessor
- an offset from the start of the PRes of the direct predecessor.

See chapter 5 for more details on calculation of PRes Response Time.

Note: From the Managing Node point of view the round trip delay for a Controlled Node equals the difference of sending time of the SyncReq and the receiving time of the SyncRes minus the length of the SyncReq. By this it is possible to calculate the propagation delay from the Managing Node to a Controlled Node.

Note: The response latency of the SyncRes is indicated in the SyncRes frame itself. If the sending of the response frame is triggered by hardware (auto-response mechanism), the latency is equal to the interframe gap of Ethernet.

3.2 Timestamps

Every Controlled Node shall record the receive-times of the SyncReq and SyncRes frames with timestamps.
On the Managing Node the send-time of the SyncReq frame and the receive-time of the SyncRes frame shall be recorded.

All time stamps shall be captured after the preamble and the start-frame-delimiter, i.e. just before the first bit of the destination MAC address.

3.3 SyncReq

The SyncReq is a special SoA frame. The service ID 6 shall be used.

To simplify configuration during bootup, the SyncReq contains PResChaining configuration data for the dedicated node.

The SyncReq frame is received by the addressed node and all other nodes in the network. All nodes memorize the receive-time and the node number of the addressed node. The addressed node replies with its SyncRes frame without delay.

Octet Offset	Description
0..5	Destination MAC Adresse
6..11	Source MAC Adresse
12..13	EtherType
14	MessageType
15	Destination
16	Source
17	NMTStatus
18	EA/ER
19	reserved
20	RequestedServiceID = 6
21	RequestedServiceTarget
22	EPLVersion
23	**reserved**
24..27	**SyncControl**
28..31	**PResTimeFirst**
32..35	**PResTimeSecond**
36..39	**SyncMNDelayFirst**
40..43	**SyncMNDelaySecond**
44..47	**PResFallBackTimeout**
48..53	**DestMacAddress**

Tab. 2 SyncReq frame

3.3.1 SyncControl (UNSIGNED32)

Bit field:

Bit Offset	Name	Description
0	PResTimeFirstValid	The parameter PResTimeFirst (UNSIGNED32) is valid. (1…active)
1	PResTimeSecondValid	The parameter PResTimeSecond (UNSIGNED32) is valid. (1…active) This bit is only set if ring redundancy is active.
2	SyncMNDelayFirstValid	The parameter SyncMNDelayFirst (UNSIGNED32) is valid. (1…active)
3	SyncMNDelaySecondValid	The parameter SyncMNDelaySecond (UNSIGNED32) is valid. (1…active) This bit is only set if ring redundancy is active.
4	PResFallBackTimeoutValid	The parameter PResFallBackTimeout (UNSIGNED32) is valid. (1…active)
5	DestMacAddressValid	The parameter DestMacAddress is valid. (1…active)
6..29	reserved	
30	PResModeReset	If this bit is set, the PResMode shall be PReq/PRes, object PResMode_U8 is set to 0 and the node shall only send its PRes triggered by the reception of the PReq. (1…active) If the bits PResModeSet and PResModeReset are set at the same time, PResModeReset overrules PResModeSet.
31	PResModeSet	If this bit is set, the PResMode shall be set to PResChaining, object PResMode_U8 is set to 1 and the node shall send its time triggered PRes in the states NMT_CS_PRE_OPERATIONAL_2 and onwards. (1…active) If the bits PResModeSet and PResModeReset are set at the same time, PResModeReset overrules PResModeSet.

Tab. 3 SyncControl bit field

3.3.2 PResTimeFirst (UNSIGNED32)

This parameter contains the PRes Response Time [ns] starting from the end of the PResMN. (see ch. 2.2)

In case of ring redundancy [2] PResTimeFirst is only valid for the primary direction of communication.

The parameter is valid and written to PResTimeFirst_U32 if the bit PResTimeFirstValid in SyncControl is set.

The new PResTimeFirst is activated immediately. It is valid from the first PRes onwards that is sent after the SyncReq.

3.3.3 PResTimeSecond (UNSIGNED32)

In case of ring redundancy [2] this parameter contains the PRes Response Time [ns] for the secondary direction of communication starting from the end of the PResMN. Otherwise this parameter shall be ignored.

The parameter is valid and written to PResTimeSecond_U32 if the bit PResTimeSecondValid in SyncControl is set.

The new PResTimeSecond is activated immediately. It is valid from the first PRes onwards that is sent after the SyncReq.

3.3.4 SyncMNDelayFirst (UNSIGNED32)

This parameter contains the propagation delay [ns] between the Managing Node and the Controlled Node.

In case of ring redundancy [2] SyncMNDelayFirst is only valid for the primary direction of communication.

The parameter is valid and written to SyncMNDelayFirst_U32 if the bit SyncMNDelayFirstValid in SyncControl is set.

3.3.5 SyncMNDelaySecond (UNSIGNED32)

In case of ring redundancy [2] this parameter contains the propagation delay [ns] between the Managing Node and the Controlled Node for the secondary direction of communication. Otherwise this parameter shall be ignored.

The parameter is valid and written to SyncMNDelaySecond_U32 if the bit SyncMNDelaySecondValid in SyncControl is set.

3.3.6 PResFallBackTimeout (UNSIGNED32)

In NMT_CN_PRE_OPERATIONAL_2 the node is not able to monitor the cycle time. The actual cycle time might not be configured yet.

The PResFallBackTimeout applies as soon as the node is in PResChaining mode. If the SoC is not received for the time of PResFallBackTimeout [ns], the PResChaining mode is deactivated. The node operates in PReq/PRes mode again.

The PResFallBackTimeout shall be set before or at the time of activation of PResMode.

3.3.7 DestMacAddress

This parameter holds the MAC address of the node the SyncReq is sent to.

Note: If the same node number is assigned to several nodes by error, all of them would accept the new settings and start sending PRes frames. This may result in collisions on the network. To avoid such problems it is recommended to use the DestMacAddress to addess only one single node.

A Controlled Node shall evaluate the parameter if the bit DestMacAddressValid in SyncControl is set. If the DestMacAddress is equal to the own MAC address the SyncReq shall be accepted. Otherwise the SyncReq shall be rejected and handled like a SoA frame not addressed to this node.

3.4 SyncRes

The SyncRes is an ASnd frame. The service ID 6 shall be used.

To simplify configuration during bootup, the SyncRes contains PResChaining configuration data of the node.

The SyncRes frame is received by all nodes supporting PResChaining. All these nodes shall check if the destination node number of the SyncReq matches the source node number of the SyncRes frame. If so the time difference and the node number are stored in the SyncRes frame of each node to be sent after receiving a SyncReq. If the node number inside the received SyncReq and SyncRes frames do not match or no SyncReq/SyncRes frames have ever been received before the values are set to 0. The time difference may be stored in the object NMT_RelativeLatencyDiff_AU32.

A plausibility check is necessary to detect wrong time differences. E.g. If a SyncReq is sent to the same node twice and a listening node misses the first SyncRes and the second SyncReq it would measure a time difference longer than the cycle time.

Octet Offset	Description
0..5	Destination MAC Adresse
6..11	Source MAC Adresse
12..13	EtherType
14	MessageType
15	Destination = C_ADR_MN_DEF_NODE_ID (0xF0)
16	Source
17	ServiceID = 6
18..19	**reserved**
20..23	**SyncStatus**
24..27	**Latency**
28..31	**SyncNodeNumber**
32..35	**SyncDelay**
36..39	**PResTimeFirst**
40..43	**PResTimeSecond**

Tab. 4 SyncRes frame

3.4.1 SyncStatus (UNSIGNED32)

Bit field:

Bit Offset	Name	Description
0	PResTimeFirstValid	The parameter PResTimeFirst (UNSIGNED32) is valid. (1…active) Attention: The value indication is delayed because the SyncRes is sent by auto-response.
1	PResTimeSecondValid	The parameter PResTimeSecond (UNSIGNED32) is valid. (1…active) Attention: The value indication is delayed because the SyncRes is sent by auto-response.
2..30	reserved	
31	PResModeStatus	This bit is set, if the PResMode is PResChaining and the node sends its time triggered PRes in the states NMT_CS_PRE_OPERATIONAL_2 and onwards. (1…active) Attention: The value indication is delayed because the SyncRes is sent by auto-response.

Tab. 5 SyncStatus bit field

3.4.2 Latency (UNSIGNED32)

This parameter contains the PollResponse latency in [ns]. The value is constant.

Note: If the PollResponse frame is sent by an auto-response mechanism, the latency is equal to the interframe gap of Ethernet.

3.4.3 SyncNodeNumber (UNSIGNED32)

SyncNodeNumber contains the node number received last inside the SyncReq/SyncRes frames.

3.4.4 SyncDelay (UNSIGNED32)

The SyncDelay contains the time difference between the end of receiving the SyncReq and the beginning of receiving the SyncRes in [ns].

3.4.5 PResTimeFirst (UNSIGNED32)

This parameter holds the current value of the PRes Response Time [ns] i.e. object PResTimeFirst_U32.

In case of ring redundancy [2] PResTimeFirst is only valid for the primary direction of communication.

If the value is changed by the SyncReq, the SyncRes immediately following the SyncReq does not yet contain the new value due to the auto-response mechanism. However, the next SyncRes frame shall contain the new value.

3.4.6 PResTimeSecond (UNSIGNED32)

In case of ring redundancy [2] this parameter holds the current value of the PRes Response Time [ns] i.e. PResTimeSecond_U32 for the secondary direction of communication. Otherwise this parameter shall be ignored.

If the value is changed by the SyncReq, the SyncRes immediately following the SyncReq does not yet contain the new value due to the auto-response mechanism. However, the next SyncRes frame shall contain the new value.

4 General operation procedures

4.1 Bootup of node

For PResChaining nodes some parameters have to be configured additionally.

- PResMode
  PReq/PRes or PResChaining (i.e. time triggered)
- PRes Response Time
  in [ns]. 32 bit = 4.29 s
- PResFallBackTimeout (UNSIGNED32)

The initial PResMode of a Controlled Node after a reset shall be PReq/PRes mode.

The Managing Node calculates the PRes Response Time and configures the PResChaining nodes with the help of SyncReq frames.

The PRes Response Time is the sum of

- The duration of all PRes frames of the predecessor nodes in the states NMT_CS_PRE_OPERATIONAL_2, NMT_CS_READY_TO_OPERATE and NMT_CS_ OPERATIONAL except the direct predecessor of this node plus
- The relative propagation times between all predecessor nodes.

See chapter 5 for more details on calculation of PRes Response Time.

If the relative propagation delay to the predecessor of a node is unknown the Managing Node has to send a SyncReq to the predecessor. After that the Managing Node sends a SyncReq to the node that needs to be inserted in the network. The relative propagation delay is part of the SyncRes frame from the new node.

Two cases have to be distinguished for configuration:

1. There is no successor node
    - The node may be inserted immediately.
2. There are one or more successor nodes
    - All PResChaining successor nodes need to be shifted backwards in the cycle.
    - This shall be done without loss of any PRes frame. The switch over of the PRes Response Times shall be done from one cycle to the next one.
    - So the shifting back shall start with the last node.

If a new node is inserted in the network the timeout for the reception of all PResChaining frames needs to be recalculated too.

4.2 Loss of a node

The Managing Node is aware of the point in time when the last PRes frame should have been received. If one or more frames are missing the dedicated loss of PRes error counters are incremented. If an error counter exceeds its threshold value the error handling is performed.

If a node, that has been lost previously, is inserted in the network again, the affected propagation delays shall be measured and the PRes Response Time shall be recalculated again too, because the topology of the network might have been changed.

4.3 Reset of a node

The PResMode of the Controlled Node is set to PReq/Pres mode by every internal or external reset performed on the node i.e. the CN stops sending time triggered PRes frames.

It is recommended to switch off PResChaining Mode of a Controlled Node by sending a dedicated SyncReq frame before sending a reset command to this node. This is to make sure the Controlled Node stops sending PRes frames in case the reset command gets lost or something similar happens.

4.4 Auto switch off of PResMode

The Controlled Node stops sending time triggered PRes frames if

- a reset is performed (see 4.3),
- the PResFallBackTimeout (UNSIGNED32) expires in NMT_CS_PRE_OPERATIONAL_2 ,
- it falls back to NMT_CS_PRE_OPERATIONAL_1 due to another error condition or
- it enters state NMT_CS_STOPPED.

After automatically switching off PResMode it shall not be switched on again by the node itself autonomously.

5 PRes Response Time

5.1 General

The PRes response time shall be calculated in a way that the start of sending the frame is during reception of the predecessor PRes frame or short afterwards.

The case that a PRes frame overtakes its predecessor frame, because of inaccurate measurements of the propagation delay, shall be avoided.

5.2 Inaccuracy in measurements

Due to the jitter of the hubs (max. 40 ns) the measurement of the propagation delay is not accurate.

In worst case the jitter is on one extreme for SyncReq and on the other extreme for SyncRes. Then the error equals 2 x #hops x 40ns. (#hops is the number of hubs between a node and its predecessor.)

5.3 Calculation of the PRes Response Time

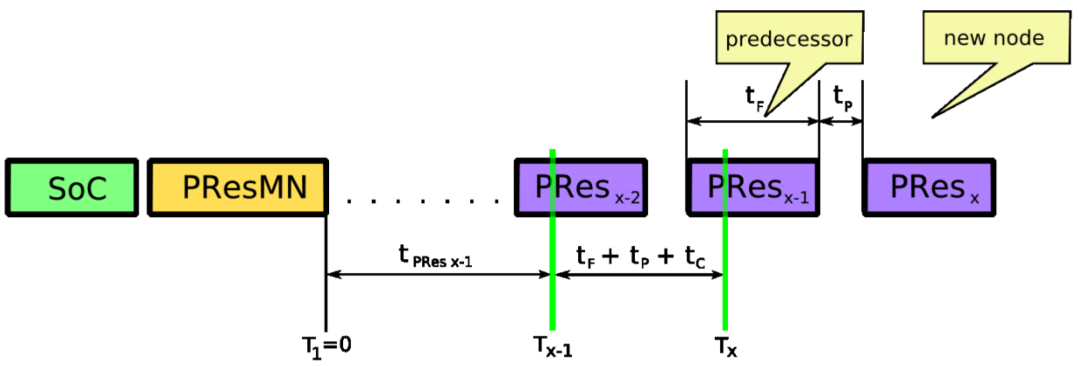

Fig. 5. Calculation of the PRes Response Time

Abbr.	Description
t_F	Frame length [time] of the PRes of the predecessor node
t_P	Relative propagation delay from the predecessor node to the new node
t_C	Additional correction time This time might be necessary to make sure that T_x is during the reception of the PRes of the predecessor node.
T_1	Timestamp at the end of receiving the PResMN frame This point in time is the trigger for sending the PRes frame for the very first node.
T_{x-1}	Timestamp of the send trigger for the PRes frame of the predecessor node.
T_x	Timestamp of the send trigger for the PRes frame of the new node.

Tab. 6 Variables for calculating the PRes Response time

$$T_x = T_{x-1} + t_F + t_P + t_C$$
$$T_1 = 0$$

Tab. 7 Formula for PRes Response Time

The first node sends its PRes immediately after receiving the PResMN frame, thus T_1 equals 0.

The additional correction time t_C might be implementation specific.

The end time of PRes Chaining i.e. the timeout for receiving the last PRes on the Managing Node may be calculated by this formula too. Thereto the predecessor node is the last node and t_P is the relative propagation delay from the last node to the Managing Node. However the correction time t_C is different to make sure that the end time is after receiving the last frame.

6 PDO Mapping

PDO mapping for PResChaining nodes is different to PReq/PRes nodes.
Output data shall be mapped into the PRes of the Managing Node instead of the PReq frame.
The Controlled Node gets its output data from the PRes of the Managing Node like usual cross-traffic.

7 Error Signaling

Because no PReq frame is sent to PResChaining nodes the StatusRequest frame shall be used for the error signalling.

However it is not necessary to poll the status continuously. As the EN flag of the PRes frame shall still be evaluated, a StatusRequest frame shall only be sent if the EN flag toggles to get the StatusResponse and once more to acknowledge the reception of the StatusResponse frame.

During initialisation of the error signalling StatusRequest frames have to be sent too.

8 Sequence of nodes in the cycle

The order of the PResChaining nodes in combination with the topology has an effect on the performance of the POWERLINK network, i.e. the minimal cycle time.

Therefore the nodes should be wired in the order of ascending or descending node numbers and send their PRes frames in one of these orders too. Thereby it does not matter if the order of the PRes frames (ascending or descending) equals the one of the wiring.

The default order may be changed by configuring the object 1F9Ch NMT_IsochrSlotAssign_AU8.

However wiring the nodes in any other order should be avoided.

9 Additional Object Description

9.1.1 Object 1050_h: NMT_RelativeLatencyDiff_AU32

NMT_RelativeLatencyDiff_AU32 holds a list of propagation delays (cf. ch. 3) from the other nodes to this node.

The object is for diagnosis only.

Index	1050_h	Object Code	ARRAY
Name	NMT_RelativeLatencyDiff_AU32		
Data Type	UNSIGNED32	Category	O

- **Sub-Index 0_h: NumberOfEntries**

Sub-Index	00_h		
Name	NumberOfEntries		
Value Range	1 .. 254	Access	ro
Default Value	254	PDO Mapping	No

- **Sub-Index 01_h .. FE_h: RelativeLatencyDiff**

Sub-Index	01_h .. FE_h		
Name	RelativeLatencyDiff		
--	--	Category	M
Value Range	UNSIGNED32	Access	ro
Default Value	0	PDO Mapping	No

Each sub-index in the array corresponds to the CN with the Node ID equal to the sub-index.

Each sub-index holds the relative propagation delay in [ns] from the respective CN to this node if measured with the help of the SyncReq/SyncRes frames. Otherwise the value of the sub-index is 0.

The object holds the propagation delay only. This time does not include the the SyncReq or SyncRes time.

9.1.2 Object $1F82_h$: NMT_FeatureFlags_U32

The Feature Flags indicate communication profile specific properties of the device given by its design. The object shall be setup by the device firmware during system initialisation.

Additional bit used:

Octet	Bit	Name	TRUE	FALSE
2	18	PResChaining	Device supports PResChaining	Device does not support PResChaining
			D_DLL_CNPResChaining_BOOL resp. D_DLL_MNPResChaining_BOOL	

Tab. 8 NMT_FeatureFlags_U32 additional bit interpretation

9.1.3 Object $1F81_h$: NMT_NodeAssignment_AU32

This object assigns nodes to the NMT Master (MN). On the CN the object is conditional. See [1] for more details.

Each sub-index in the array corresponds to the node with the node ID equal to the sub-index.

The object should be set by the system configuration.

Additional bit used:

Index	$1F81_h$	Object Code	ARRAY
Name	NMT_NodeAssignment_AU32		
Data Type	UNSIGNED32	Category	MN: M CN: Cond

- **Sub-Index 01_h .. FE_h: NodeAssignment**

Sub-Index	01_h .. FE_h		
Name	NodeAssignment		
--	--	Category	M
Value Range	Bit field, see below	Access	rw, valid on reset
Default Value	0	PDO Mapping	No

Octet	Bit	Value	Description	Property	Evaluate
2	14	0_b	Conventional PReq/PRes node	CN	MN
		1_b	PResChaining node (No PReq is sent to this node)		

Tab. 9 NMT_NodeAssignment_AU32 additional bit interpretation

9.1.4 Object $1F98_h$: NMT_CycleTiming_REC

NMT_CycleTiming_REC provides node specific timing parameters, that influence the POWERLINK cycle timing.

Additional subindices used on the CN only:

Index	$1F98_h$	Object Code	RECORD
Name	NMT_CycleTiming_REC		
Data Type	NMT_CycleTiming_TYPE	Category	M

- **Sub-Index 00_h: NumberOfEntries**

Sub-Index	00_h		
Name	NumberOfEntries		
Value Range	9 .. 14	Access	const
Default Value	-	PDO Mapping	No

- **Sub-Index $0A_h$: PResMode_U8**

Sub-Index	$0A_h$		
Name	PResMode_U8		
Data Type	UNSIGNED8	Category	MN: no, CN: O
Value Range	0 … PReq/PRes 1 … PResChaining	Access	ro
Default Value	0	PDO Mapping	No

This value is set by the dedicated bit from SyncControl (UNSIGNED32) of the SyncReq frame and may be read for diagnosis.

- **Sub-Index 0Bh: PResTimeFirst_U32**

Sub-Index	0Bh		
Name	PResTimeFirst_U32		
Data Type	UNSIGNED32	Category	MN: no, CN: O
Value Range	UNSIGNED32	Access	ro
Default Value	0	PDO Mapping	No

This value [ns] is set by the field PResTimeFirst (UNSIGNED32) of the SyncReq frame and may be read for diagnosis.

If the PResMode is PResChaining and the SoC has been received from the same port than the IdentRequest frame, the CNs sends its PRes frame with a delay of PResTimeFirst_U32 [ns]. If the PResMode is PReq/PRes this value is 0, i.e. is not valid.

- **Sub-Index 0Ch: PResTimeSecond_U32**

Sub-Index	0Ch		
Name	PResTimeSecond_U32		
Data Type	UNSIGNED32	Category	MN: no, CN: O
Value Range	UNSIGNED32	Access	ro
Default Value	0	PDO Mapping	No

This value [ns] is set by the field PResTimeSecond (UNSIGNED32) of the SyncReq frame and may be read for diagnosis.

If the PResMode is PResChaining and the SoC has not been received from the same port than the IdentRequest frame, the CNs sends its PRes frame with a delay of PResTimeSecond_U32 [ns].

This sub-index is 0, i.e. is not valid, if the PResMode is PReq/PRes or ring redundancy is not active.

- **Sub-Index 0Dh: SyncMNDelayFirst_U32**

Sub-Index	0Dh		
Name	SyncMNDelayFirst_U32		
Data Type	UNSIGNED32	Category	MN: no, CN: O
Value Range	UNSIGNED32	Access	ro
Default Value	0	PDO Mapping	No

This value [ns] is set by the field SyncMNDelayFirst (UNSIGNED32) of the SyncReq frame and may be read for diagnosis.

If the PResMode is PResChaining, the sub-index holds the propagation time between the MN and this CN in case the SoC has been received from the same port than the IdentRequest frame. For the other case the propagation delay is hold in SyncMNDelaySecond_U32.

The CN may use this value for a more precise synchronization, i.e. to compensate the skew.

If the PResMode is PReq/PRes this sub-index is 0, i.e. is not valid.

- **Sub-Index 0Eh: SyncMNDelaySecond_U32**

Sub-Index	0Eh		
Name	SyncMNDelaySecond_U32		
Data Type	UNSIGNED32	Category	MN: no, CN: O
Value Range	UNSIGNED32	Access	ro
Default Value	0	PDO Mapping	No

This value [ns] is set by the field SyncMNDelaySecond (UNSIGNED32) of the SyncReq frame and may be read for diagnosis.

If the PResMode is PResChaining, the sub-index holds the propagation time between the MN and this CN in case the SoC has not been received from the same port than the IdentRequest frame. For the other case the propagation delay is hold in SyncMNDelayFirst_U32.

The CN may use this value for a more precise synchronization, i.e. to compensate the skew.

This sub-index is 0, i.e. is not valid, if the PResMode is PReq/PRes or ring redundancy is not active.

App. 1 Device Description Entries (normative)

Additional device description entries:

Name	Description	Type	Category		Default	
			MN	CN	MN	CN
D_DLL_CNPResChaining_BOOL	Ability of a Controlled Node to perform PResChaining functions	BOOLEAN	-	M	-	N
D_DLL_MNPResChaining_BOOL	Ability of a Managing Node to perform PResChaining functions	BOOLEAN	M	-	N	-

end-of-file