Document Title	SRS_IPDUMultiplexer: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1 SRS_IPDUMultiplexer

1.1 Specification Item SRS_IpduM_02820

Trace References:

RS_BRF_01716

Content:

Type:	Valid
Description:	One or more I-PDUs shall be mappable to a Container-PDU. If an I-PDU is mapped to a Container, it shall be subject to the Multi-PDU-to-Container Mapping.
Rationale:	Mapping multiple PDUs to one frame reduces bus load by improving usage of enhanced bandwidth.
Applies to:	
Use Case:	Efficient transmission of small PDUs of high bandwidth busses
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==
  Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

===
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the

same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.2 Specification Item SRS_IpduM_02825

Trace References:

RS_BRF_01716

Content:

Type:	Draft
Description:	Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale:	Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Applies to:	
Use Case:	Efficient transmission of small PDUs of high bandwidth busses
Supporting Material:	–

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:

- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the

IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==

Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1