Document Title	PRS_SOMEIPServiceDiscoveryProtocol: Complete Change Documentation 1.4.0 - 1.5.0
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	885

Document Status	Final
Part of AUTOSAR Standard	Foundation
Part of Standard Release	1.5.0

Table of Contents

1 PRS_SOMEIPServiceDiscoveryProtocol

1.1 Specification Item PRS_SOMEIPSD_00124

Trace References:

RS_SOMEIPSD_00019

Content:

For error handling of incoming SOME/IP-SD messages, Execute the checks described in (PRS_SOMEIPSD_00125, PRS_SOMEIPSD_00126, PRS_SOMEIPSD_00127, PRS_SOMEIPSD_00128, PRS_SOMEIPSD_00129, PRS_SOMEIPSD_00803, PRS_SOMEIPSD_00130, PRS_SOMEIPSD_00131, PRS_SOMEIPSD_00132). If at least one of these checks fails, you need to:

- Answer with a Subscribe Eventgroup NACK, if the original entry was a Subscribe Eventgroup entry
- Ignore, if the original entry was not a Subscribe Eventgroup entry

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK en-

try, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.2 Specification Item PRS_SOMEIPSD_00125

Trace References:

RS_SOMEIPSD_00019

Content:

Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be

discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for

a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.3 Specification Item PRS_SOMEIPSD_00126

Trace References:

RS_SOMEIPSD_00019

Content:

Check if If the Service ID is known of a received entry is not known, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
  bullet 2 and 3
  * if number of opt1 equals 0, the Index 1st options also equals 0
  * if number of opt2 equals 0, the Index 2nd options also equals 0
  –Last change on issue 79206 comment 18–

  Agreed solution:

  ~[PRS_SOMEIPSD_00273]
  add after Type (last bullet point)
  + * Discardable Flag [1 bit]: Specifies if the option can be discarded.
  + * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.4 Specification Item PRS_SOMEIPSD_00127

Trace References:

RS_SOMEIPSD_00019

Content:

Check if If the Instance ID of this Service ID is known a received entry is not known, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array

and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).

* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.5 Specification Item PRS_SOMEIPSD_00128

Trace References:

RS_SOMEIPSD_00019

Content:

Check if If the Major Version of this Service Instance is known a received entry is not known, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or

[PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.6 Specification Item PRS_SOMEIPSD_00129

Trace References:

RS_SOMEIPSD_00019

Content:

Check if If the Eventgroup ID of the Service Instance with Major Version is known (only applicable for eventgroup entries) a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for

entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid

size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.7 Specification Item PRS_SOMEIPSD_00130

Trace References:

RS_SOMEIPSD_00019

Content:

Check if the referenced Options exist in the options array and are syntactically okof each received entry:

- Length of Options Array is consistent The referenced options exist.

- if number of opt1 equals 0, the Index 1st options also equals 0 The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- if number of opt2 equals 0, the Index 2nd options also equals 0 The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- Option Type is known There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- Option Length is consistent The Type of the referenced Option is known or the discardable flag is set to 1.
- Option The Type of the referenced Option is valid for entry allowed for the entry PRS_SOMEIPSD_00583 or discardable flag is set to 1.
- Endpoint Options with The Length of the referenced Option is consistent to the Type of the Option.
- An Endpoint Option has a valid L4-Protocol field.
- The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced

from an offer is not allowed and invalid.)
Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received

Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.8 Specification Item PRS_SOMEIPSD_00231

Trace References:

RS_SOMEIPSD_00019

Content:

Options that are referenced by an entry shall be ignored if:

- The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
- The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
  bullet 2 and 3
  * if number of opt1 equals 0, the Index 1st options also equals 0
  * if number of opt2 equals 0, the Index 2nd options also equals 0
  –Last change on issue 79206 comment 18–

  Agreed solution:

  ~[PRS_SOMEIPSD_00273]
  add after Type (last bullet point)
  + * Discardable Flag [1 bit]: Specifies if the option can be discarded.
  + * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.9 Specification Item PRS_SOMEIPSD_00232

Trace References:

RS_SOMEIPSD_00019

Content:

If the checks in PRS_SOMEIPSD_00130 fail for a received Find entry, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array

and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).

* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.10 Specification Item PRS_SOMEIPSD_00233

Trace References:

RS_SOMEIPSD_00019

Content:

If the checks in PRS_SOMEIPSD_00130 fail for a received Offer entry, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or

[PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.11 Specification Item PRS_SOMEIPSD_00234

Trace References:

RS_SOMEIPSD_00019

Content:

If the checks in PRS_SOMEIPSD_00130, PRS_SOMEIPSD_00131, or PRS_SOMEIPSD_00132 fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for

entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid

size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.12 Specification Item PRS_SOMEIPSD_00235

Trace References:

RS_SOMEIPSD_00019

Content:

If the checks in PRS_SOMEIPSD_00130 or PRS_SOMEIPSD_00132 fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
  bullet 2 and 3
  * if number of opt1 equals 0, the Index 1st options also equals 0
  * if number of opt2 equals 0, the Index 2nd options also equals 0
  –Last change on issue 79206 comment 18–

  Agreed solution:

  ~[PRS_SOMEIPSD_00273]
  add after Type (last bullet point)
  + * Discardable Flag [1 bit]: Specifies if the option can be discarded.
  + * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.13 Specification Item PRS_SOMEIPSD_00273

Trace References:

RS_SOMEIPSD_00006

Content:

In order to identify the option type every option shall start with:

- Length [uint16]: Specifies the length of the option in Bytes.
- Type [uint8]: Specifying the type of the option.
- Discardable Flag [1 bit]: Specifies if the option can be discarded.
- Bit 1 to bit 7 are reserved and shall be 0.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received

entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be

ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.14 Specification Item PRS_SOMEIPSD_00275

Trace References:

RS_SOMEIPSD_00006

Content:

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in

PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.15 Specification Item PRS_SOMEIPSD_00276

Trace References:

RS_SOMEIPSD_00006

Content:

The format of the Configuration Option shall be as follows:

- Length [uint16]: Shall be set to the total number of bytes occupied by the configuration option, excluding the 16 bit length field and the 8 bit type flag.
- Type [uint8]: Shall be set to 0x01.
- Reserved Discardable Flag [uint81 bit]: Shall be set to 0x00. 1 if the Option can be discarded by the receiver.
- Bit 1 to bit 7 are reserved and shall be 0.
- ConfigurationString [dyn length]: Shall carry the configuration string.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
  bullet 2 and 3
  * if number of opt1 equals 0, the Index 1st options also equals 0
  * if number of opt2 equals 0, the Index 2nd options also equals 0
  –Last change on issue 79206 comment 18–

  Agreed solution:

  ~[PRS_SOMEIPSD_00273]
  add after Type (last bullet point)
  + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.

* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.16 Specification Item PRS_SOMEIPSD_00307

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The Format of the IPv4 Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0009.
- Type [uint8]: Shall be set to 0x04.
- Reserved Discardable Flag [uint81 bit]: Shall be set to 0x00. 0.
- Bit 1 to bit 7 are reserved and shall be 0.
- IPv4-Address [uint32]: Shall transport the unicast IP-Address as four Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol (ISO/OSI layer 4) based on the IANA/IETF types (0x06: TCP, 0x11: UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the port of the layer 4 protocol.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced

from an offer is not allowed and invalid.)
Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received

Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.17 Specification Item PRS_SOMEIPSD_00315

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The Format of the IPv6 Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0015.
- Type [uint8]: Shall be set to 0x06.
- Reserved Discardable Flag [uint81 bit]: Shall be set to 0x00. 0.
- Bit 1 to bit 7 are reserved and shall be 0.
- IPv6-Address [uint128]: Shall transport the unicast IP-Address as 16 Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol (ISO/OSI layer 4) based on the IANA/IETF types (0x06: TCP, 0x11: UDP).

- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the transport layer port(e.g. 30490).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
  bullet 2 and 3
  * if number of opt1 equals 0, the Index 1st options also equals 0
  * if number of opt2 equals 0, the Index 2nd options also equals 0
  –Last change on issue 79206 comment 18–

  Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to

PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.18 Specification Item PRS_SOMEIPSD_00326

Trace References:

RS_SOMEIPSD_00003

Content:

The Format of the IPv4 Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0009.
- Type [uint8]: Shall be set to 0x14.
- Reserved Discardable Flag [uint81 bit]: Shall be set to 0x00. 0.
- Bit 1 to bit 7 are reserved and shall be 0.
- IPv4-Address [uint32]: Shall transport the multicast IP-Address as four Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol (ISO/OSI layer 4) based on the IANA/IETF types (0x11: UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the port of the layer 4 protocol.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid

as descibed in PRS.
If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known,

the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for

a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.19 Specification Item PRS_SOMEIPSD_00333

Trace References:

RS_SOMEIPSD_00003

Content:

The Format of the IPv6 Multicast Option shall be as follows:

- Length [uint16]: Shall be set to 0x0015.
- Type [uint8]: Shall be set to 0x16.
- Reserved Discardable Flag [uint81 bit]: Shall be set to 0x00. 0.
- Bit 1 to bit 7 are reserved and shall be 0.
- IPv6-Address [uint128]: Shall transport the multicast IP-Address as 16 Bytes.

- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol (ISO/OSI layer 4) based on the IANA/IETF types (0x11: UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the port of the layer 4 protocol.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
  bullet 2 and 3
  * if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).

* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup

entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.20 Specification Item PRS_SOMEIPSD_00544

Trace References:

RS_SOMEIPSD_00011

Content:

The Format of the Load Balancing Option shall be as follows:

- Length [uint16]: Shall be set to 0x0005.
- Type [uint8]: Shall be set to 0x02.
- Reserved Discardable Flag [uint81 bit]: Shall be set to 0x00. 1 if the Option can be discarded by the receiver.
- Bit 1 to bit 7 are reserved and shall be 0.
- Priority [uint16]: Carries the Priority of this instance. Lower value means higher priority.
- Weight [uint16]: Carries the Weight of this instance. Large value means higher probability to be chosen.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.21 Specification Item PRS_SOMEIPSD_00551

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number and Port Number of the sender used for Service Discovery.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80342: [PRS_SOMEIPSD] Redundant requirements for SD endpoint option

Problem description:

During the document review one statement to be clarified was identified:

Section 4 Protocol Specification, [PRS_SOMEIP_00710].

It seems to be a redundant copy of [PRS_SOMEIP_00653], thus it should be removed.

A comprehensive feedback would be appreciated in order to go ahead with the document release.

Agreed solution:

- [PRS_SOMEIPSD_00710]
- [PRS_SOMEIPSD_00653]

~[PRS_SOMEIPSD_00558]
- The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number.
+ The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

~[PRS_SOMEIPSD_00551]
- The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number.
+ The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.
–Last change on issue 80342 comment 9–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.22 Specification Item PRS_SOMEIPSD_00552

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The Format of the IPv4 SD Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0015.
- Type [uint8]: Shall be set to 0x24.
- Reserved Discardable Flag [uint81 bit]: Shall be set to 0x00. 0.
- Bit 1 to bit 7 are reserved and shall be 0.
- IPv4-Address [uint32]: Shall transport the unicast IP-Address of SOME/IP-SD as four Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol of SOME/IP-SD (currently: 0x11 UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the transport layer port of SOME/IP-SD (currently: 30490).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and

ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.23 Specification Item PRS_SOMEIPSD_00558

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number and Port Number of the sender used for Service Discovery.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80342: [PRS_SOMEIPSD] Redundant requirements for SD endpoint option

  Problem description:

  During the document review one statement to be clarified was identified:

  Section 4 Protocol Specification, [PRS_SOMEIP_00710].

  It seems to be a redundant copy of [PRS_SOMEIP_00653], thus it should be removed.

  A comprehensive feedback would be appreciated in order to go ahead with the document release.

Agreed solution:

- [PRS_SOMEIPSD_00710]
- [PRS_SOMEIPSD_00653]

~[PRS_SOMEIPSD_00558]
- The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number.
+ The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

~[PRS_SOMEIPSD_00551]
- The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number.
+ The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.
–Last change on issue 80342 comment 9–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.24 Specification Item PRS_SOMEIPSD_00559

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The Format of the IPv6 SD Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0015.
- Type [uint8]: Shall be set to 0x26.
- Reserved Discardable Flag [uint81 bit]: Shall be set to 0x00. 0.
- Bit 1 to bit 7 are reserved and shall be 0.
- IPv6-Address [uint128]: Shall transport the unicast IP-Address of SOME/IP-SD as 16 Bytes.
- Reserved [uint8]: Shall be set to 0x00.

- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol of SOME/IP-SD (currently: 0x11 UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the transport layer port of SOME/IP-SD (currently: 30490).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

  I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

  Contradicting requirements:
  [SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
  and
  PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
  bullet 2 and 3
  * if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).

* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup

entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.25 Specification Item PRS_SOMEIPSD_00653

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The IPv6 SD Endpoint Option shall only be included if the SOME/IP-SD message is transported over IPv6.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80342: [PRS_SOMEIPSD] Redundant requirements for SD endpoint option

  Problem description:

  During the document review one statement to be clarified was identified:

  Section 4 Protocol Specification, [PRS_SOMEIP_00710].

  It seems to be a redundant copy of [PRS_SOMEIP_00653], thus it should be removed.

  A comprehensive feedback would be appreciated in order to go ahead with the document release.

  Agreed solution:

  - [PRS_SOMEIPSD_00710]
  - [PRS_SOMEIPSD_00653]

  ~[PRS_SOMEIPSD_00558]
  - The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number.
  + The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

~[PRS_SOMEIPSD_00551]
- The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number.
+ The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.
–Last change on issue 80342 comment 9–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.26 Specification Item PRS_SOMEIPSD_00710

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The IPv6 SD Endpoint Option shall only be included if the SOME/IP-SD message is transported over IPv6.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80342: [PRS_SOMEIPSD] Redundant requirements for SD endpoint option

  Problem description:

  During the document review one statement to be clarified was identified:

  Section 4 Protocol Specification, [PRS_SOMEIP_00710].

  It seems to be a redundant copy of [PRS_SOMEIP_00653], thus it should be removed.

  A comprehensive feedback would be appreciated in order to go ahead with the document release.

  Agreed solution:

  - [PRS_SOMEIPSD_00710]
  - [PRS_SOMEIPSD_00653]

~[PRS_SOMEIPSD_00558]
- The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number.
+ The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

~[PRS_SOMEIPSD_00551]
- The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number.
+ The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.
–Last change on issue 80342 comment 9–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.27 Specification Item PRS_SOMEIPSD_00711

Trace References:

RS_SOMEIPSD_00011

Content:

When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority that also matches client specific criteria.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79832: [PRS SD] Clarify load balancing option usage

  Problem description:

  The current SOME/IP Service Discovery Protocol Specification does not specify how to proceed if a service using load balancing options is already in use (i.e. ongoing method calls and subscribed event groups) and an offer service with higher priority is received.

  How shall a service be selected if one service instance uses a load balancing option and an other instance does not.

Agreed solution:

+[PRS_SOMEIPSD_00xxx]
When looking for a specific service instances of a service (Service Instance set to any value other than 0xFFFF), the evaluation of the Load Balancing Option does not apply.

~[PRS_SOMEIPSD_00711]
- When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority.
+ When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority that also matches client specific criteria.
Note: Client specific criteria may be applied by the client application when choosing one of the offered service instances. They are not defined in this specification, and could e.g. restrict the range of appropriate instance IDs.

~[PRS_SOMEIPSD_00713]
- In case if there is no Load Balancing Option available and several Service instance are offered, then the application logic shall perform the selection of the Service instance based on use-case specific criteria.
+ If an Offer Service entry references no Load Balancing option and several service instances are offered, the client shall handle the service instances without Load Balancing option as though they had the lowest priority.
–Last change on issue 79832 comment 45–

BW-C-Level:

Application	Specification	Bus
4	4	4

1.28 Specification Item PRS_SOMEIPSD_00713

Trace References:

RS_SOMEIPSD_00011

Content:

In case if there is If an Offer Service entry references no Load Balancing Option avilable and several Service instance option and several service instances are offered, then the application logic shall perform the selection of the Service instance based on use-case specific criteriathe client shall handle the service instances without Load Balancing option as though they had the lowest priority.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79832: [PRS SD] Clarify load balancing option usage

  Problem description:

  The current SOME/IP Service Discovery Protocol Specification does not specify how to proceed if a service using load balancing options is already in use (i.e. ongoing method calls and subscribed event groups) and an offer service with higher priority is received.

  How shall a service be selected if one service instance uses a load balancing option and an other instance does not.

  Agreed solution:

  +[PRS_SOMEIPSD_00xxx]
  When looking for a specific service instances of a service (Service Instance set to any value other than 0xFFFF), the evaluation of the Load Balancing Option does not apply.

  ~[PRS_SOMEIPSD_00711]
  - When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority.
  + When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority that also matches client specific criteria.
  Note: Client specific criteria may be applied by the client application when choosing one of the offered service instances. They are not defined in this specification, and could e.g. restrict the range of appropriate instance IDs.

  ~[PRS_SOMEIPSD_00713]
  - In case if there is no Load Balancing Option available and several Service instance are offered, then the application logic shall perform the selection of the Service instance based on use-case specific criteria.
  + If an Offer Service entry references no Load Balancing option and several service instances are offered, the client shall handle the service instances without Load Balancing option as though they had the lowest priority.
  –Last change on issue 79832 comment 45–

  BW-C-Level:

Application	Specification	Bus
4	4	4

1.29 Specification Item PRS_SOMEIPSD_00714

Trace References:

RS_SOMEIPSD_00011

Content:

When looking for a specific service instances of a service (Service Instance set to any value other than 0xFFFF), the evaluation of the Load Balancing Option does not apply.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79832: [PRS SD] Clarify load balancing option usage

  Problem description:

  The current SOME/IP Service Discovery Protocol Specification does not specify how to proceed if a service using load balancing options is already in use (i.e. ongoing method calls and subscribed event groups) and an offer service with higher priority is received.

  How shall a service be selected if one service instance uses a load balancing option and an other instance does not.

  Agreed solution:

  +[PRS_SOMEIPSD_00xxx]
  When looking for a specific service instances of a service (Service Instance set to any value other than 0xFFFF), the evaluation of the Load Balancing Option does not apply.

  ~[PRS_SOMEIPSD_00711]
  - When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority.
  + When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority that also matches client specific criteria.
  Note: Client specific criteria may be applied by the client application when choosing one of the offered service instances. They are not defined in this specification, and could e.g. restrict the range of appropriate instance IDs.

  ~[PRS_SOMEIPSD_00713]
  - In case if there is no Load Balancing Option available and several Service instance are offered, then the application logic shall perform the selection of the Service instance based on use-case specific criteria.

+ If an Offer Service entry references no Load Balancing option and several service instances are offered, the client shall handle the service instances without Load Balancing option as though they had the lowest priority.
–Last change on issue 79832 comment 45–

BW-C-Level:

Application	Specification	Bus
4	4	4

1.30 Specification Item PRS_SOMEIPSD_00803

Trace References:

RS_SOMEIPSD_00019

Content:

Check that If the length of the Entries Array has a valid size an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

  Problem description:

  PRS_SOMEIPSD_00130 is ambigous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

  SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"
  SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

  SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.
  If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)
  Can anyone give an exhaustive list of options that are redundant and still valid, so

that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:
[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and
PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:
bullet 2 and 3
* if number of opt1 equals 0, the Index 1st options also equals 0
* if number of opt2 equals 0, the Index 2nd options also equals 0
–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]
add after Type (last bullet point)
+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.
+ * Bit 1 to bit 7 are reserved and shall be 0.

+[PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]
The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 0.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]
- * Reserved [uint8]: Shall be set to 0x00.
+ * Discardable Flag [1 bit]: Shall be set to 1.
+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not

known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:
* The referenced options exist.
* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
* There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
* The Type of the referenced Option is known or the discardable flag is set to 1.
* The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
* The Length of the referenced Option is consistent to the Type of the Option.
* An Endpoint Option has a valid L4-Protocol field.
* The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:
Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+[PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+[PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+[PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+[PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:
* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

1.31 Specification Item PRS_SOMEIPSD_00822

Trace References:

RS_SOMEIPSD_00015

Content:

If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 0 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send no notifications/events (i.e. initial events)immediately after sending the Subscribe Eventgroup Ack.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80067: [PRS SD] Contradicting and incomplete requirements for initial data request flag on server side

  Problem description:

  PRS_SOMEIPSD_00270, PRS_SOMEIPSD_00386, PRS_SOMEIPSD_00703 and PRS_SOMEIPSD_00463 specify when the initial data request flag shall be set to 1.

  The implied semantic of the initial data request flag contradicts the specified

reaction on the server as specified in PRS_SOMEIPSD_00822.

According to PRS_SOMEIPSD_00822 the initial data shall be sent independent of the request flag, making the request flag useless.
–Last change on issue 80067 comment 1–

Agreed solution:

~PRS_SOMEIPSD_00822
- If the server receives a Subscribe Eventgroup entry with the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.
+ If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 0 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send no notifications/events (i.e. initial events).

add PRS_SOMEIPSD_00xxx after PRS_SOMEIPSD_00822
+ If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 1 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

-PRS_SOMEIPSD_00823
–Last change on issue 80067 comment 25–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.32 Specification Item PRS_SOMEIPSD_00823

Trace References:

RS_SOMEIPSD_00015

Content:

The client shall repeat the Subscribe Eventgroup entry, if it did not receive the notifications/events in a configurable timeout independent of the setting of the Explicit Initial Data Control Flag.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80067: [PRS SD] Contradicting and incomplete requirements for initial data request flag on server side

  Problem description:

  PRS_SOMEIPSD_00270, PRS_SOMEIPSD_00386, PRS_SOMEIPSD_00703 and PRS_SOMEIPSD_00463 specify when the initial data request flag shall be set to 1.

  The implied semantic of the initial data request flag contradicts the specified reaction on the server as specified in PRS_SOMEIPSD_00822.

  According to PRS_SOMEIPSD_00822 the initial data shall be sent independent of the request flag, making the request flag useless.
  –Last change on issue 80067 comment 1–

  Agreed solution:

  ~PRS_SOMEIPSD_00822
  - If the server receives a Subscribe Eventgroup entry with the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.
  + If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 0 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send no notifications/events (i.e. initial events).

  add PRS_SOMEIPSD_00xxx after PRS_SOMEIPSD_00822
  + If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 1 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

  -PRS_SOMEIPSD_00823
  –Last change on issue 80067 comment 25–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.33 Specification Item PRS_SOMEIPSD_00824

Trace References:

RS_SOMEIPSD_00015

Content:

If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 1 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80067: [PRS SD] Contradicting and incomplete requirements for initial data request flag on server side

  Problem description:

  PRS_SOMEIPSD_00270, PRS_SOMEIPSD_00386, PRS_SOMEIPSD_00703 and PRS_SOMEIPSD_00463 specify when the initial data request flag shall be set to 1.

  The implied semantic of the initial data request flag contradicts the specified reaction on the server as specified in PRS_SOMEIPSD_00822.

  According to PRS_SOMEIPSD_00822 the initial data shall be sent independent of the request flag, making the request flag useless.
  –Last change on issue 80067 comment 1–

  Agreed solution:

  ~PRS_SOMEIPSD_00822
  - If the server receives a Subscribe Eventgroup entry with the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.
  + If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 0 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send no notifications/events (i.e. initial events).

  add PRS_SOMEIPSD_00xxx after PRS_SOMEIPSD_00822
  + If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 1 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately

after sending the Subscribe Eventgroup Ack.

-PRS_SOMEIPSD_00823
–Last change on issue 80067 comment 25–

BW-C-Level:

Application	Specification	Bus
1	1	1