Solution to redmine 7100

**Tissue 1934 Subscriber with beh=on receives GOOSE with q.test=true, then subscriber changes to test mode**

March 3, 2025

The tissue states that the DUT need to re-process the last GOOSE message data values in case the Mod/Beh changes from on to test or vice versa. The DUT is not allowed to wait for the next retransmission of the GOOSE. We need to add a new test case.

| sGos24 | Verify that the DUT process GOOSE data values with quality test when changing Beh to test and on and vice versa prior to the next retransmission. |
| --- | --- |

The test procedure is extended to test Mod=blocked and test/blocked as well.

|  |  |  |
| --- | --- | --- |
| **sGos24** | **Verify that the DUT process GOOSE data values with quality test when changing Beh to test and vice versa** | [ ]  Passed[ ]  Failed[ ]  Inconclusive |
| IEC 61850-7-4 Annex APIXIT Sr5, Gs12 = ‘process as invalid’ shall be different from ‘processed as valid’ when configurable |
| Expected result1. LN.Beh = **on** 2. DUT processes both Ind1 & Ind2 values flagged quality test FALSE as ‘valid’  DUT update the values and sends a GOOSE message or Report with the changed values 3. DUT processes the Ind1 value flagged with quality test TRUE as ‘**invalid’** (PIXIT)  DUT processes the Ind2 value flagged with quality test FALSE as ‘valid’;  DUT updates the values and sends a GOOSE message or Report with the changed values, prior to the next re-transmit4. LN.Beh = **blocked** DUT processes the Ind1 value flagged quality test TRUE as ‘**invalid’** (PIXIT) DUT processes the Ind2 value flagged quality test FALSE as ‘valid’ DUT sends a GOOSE message or Report with the changed, prior to the next re-transmit5. DUT processes the Ind1 value flagged quality test TRUE as ‘**invalid’** (PIXIT) DUT processes the Ind2 value flagged quality test FALSE as ‘valid’ DUT updates the values and sends a GOOSE message or Report with the changed values6. LN.Beh = **test** DUT processes the Ind1 value flagged quality test TRUE as ‘valid’ DUT processes the Ind2 value flagged quality test FALSE as ‘valid’ DUT sends a GOOSE message or Report with the changed values both flagged quality test true, prior to the next re-transmit7. DUT processes the Ind1 value flagged quality test TRUE as ‘valid’ DUT processes the Ind2 value flagged quality test FALSE as ‘valid’ DUT updates the values and sends a GOOSE message or Report with the changed values both flagged quality test true 8. LN.Beh = **test/blocked** DUT processes the Ind1 value flagged quality test TRUE as ‘valid’ DUT processes the Ind2 value flagged quality test FALSE as ‘valid’ DUT sends a GOOSE message or Report with the changed values both flagged quality test true, prior to the next re-transmit9. DUT processes the Ind1 value flagged quality test TRUE as ‘valid’ DUT processes the Ind2 value flagged quality test FALSE as ‘valid’ DUT updates the values and sends a GOOSE message or Report with the changed values both flagged quality test true 10. LN.Beh = **on** DUT processes the Ind1 value flagged quality test TRUE as ‘**invalid’** (PIXIT) DUT processes the Ind2 value flagged quality test FALSE as ‘valid’ DUT updates the values and sends a GOOSE message or Report with the changed values, prior to the next re-transmit 11. DUT processes both Ind1&Ind2 values flagged quality test FALSE as ‘valid’ DUT updates the values and sends a GOOSE message or Report with the changed values  |
| Test descriptionTest engineer configures the DUT with the ping-pong mechanism for FCDA. One of the FCDA: Ind1.q.test will change from FALSE to TRUE to FALSE in the following steps, the Ind2.q.test will remain FALSE for the entire time. Both the ping Ind1 and Ind2 are subscribed by the DUT and copied to the pong Ind1 and Ind2.1. Force the subscriber Logical Node into Beh = on 2. SIMULATOR publishes GOOSE message with changed data values Ind1 and Ind2 flagged quality test false each3. SIMULATOR publishes GOOSE message with changed data values Ind1 flagged quality test true, Ind2 flagged quality test false 4. Force the subscriber Logical Node into Beh = blocked (when supported)5. SIMULATOR keeps publishing GOOSE messages with changed data values Ind1 flagged quality test true, Ind2 flagged as test false6 Force the subscriber Logical Node into Beh = test7. SIMULATOR keeps publishing GOOSE messages with changed data values Ind1 flagged quality test true, Ind2 flagged as test false8 Force the subscriber Logical Node into Beh = test/blocked (when supported)9. SIMULATOR keeps publishing GOOSE messages with changed data values Ind1 flagged quality test true, Ind2 flagged as test false10. Force the subscriber Logical Node into Beh = on11. SIMULATOR publishes GOOSE message with changed data values Ind1 and Ind2 flagged as test false |
| Comment |

For convenience (no change)

|  |  |  |  |
| --- | --- | --- | --- |
| Gs12 | Amd1 | Is the “processing data as invalid” configurable?When not configurable how does the subscriber “process data as invalid”? | Y/NKeep last non test value Y/NSubstitute to a configured value Y/NSet derived quality to invalid: Y/N Other: <describe> |

Typical behavior ‘process as invalid’ = set q.invalid, keep last value

DUT Beh=on; receive PingInd1.q=test and PingInd2 => send PongInd1.q = invalid; PongInd2.q=valid

DUT Beh=test; receive PingInd1.q=test and PingInd2 => send PongInd1.q = valid+test; PongInd2.q=invalid

About step 4; when Beh=test incoming data with q.test=FALSE shall be ‘processed as valid’

Table A.2

