|  |  |  |
| --- | --- | --- |
| **sGos6** | **Subscribe GOOSE with simulation parameter set** | Passed   Failed   Inconclusive |
| IEC 61850-7-1 Subclause 7.8.2  IEC 61850-7-2 Subclause 18.2.3.8  IEC 61850-8-1 Subclause 18.1  PIXIT: Gs9 | | |
| Expected result   1. LPHD.Sim.stVal = FALSE or not present 2. DUT accepts the normal GOOSE messages, LGOS.St = TRUE, LGOS.SimSt=FALSE, DUT accepts GOOSE message from Publisher 3, LGOS3.St = TRUE, LGOS3.SimSt = FALSE and does not change in the following steps. 3. DUT ignores the simulated data value change, LGOS.St=TRUE, LGOS.SimSt=FALSE 4. DUT changes LGOS.St.stVal to FALSE (and keeps LGOS.SimSt = FALSE) 5. DUT keeps LGOS.St.stVal=FALSE 6. LPHD.Sim.stVal = TRUE 7. DUT accepts the Publisher 1 GOOSE messages because no simulated GOOSE messages have been received yet, LGOS.St=TRUE, LGOS.SimSt=FALSE; state: subscription normal GOOSE as long as no simulated GOOSE received. 8. DUT changes LGOS.SimSt=TRUE (and keeps LGOS.St=TRUE); state: subscription simulated GOOSE 9. DUT accepts the simulated data value change 10. DUT changes LGOS.St to FALSE (and keeps LGOS.SimSt=TRUE); state: wait for simulated GOOSE 11. DUT ignores the normal GOOSE messages 12. DUT keeps LGOS.St=FALSE and LGOS.SimSt=TRUE and continues to accept GOOSE messages from Publisher 3 13. DUT changes LPHD.Sim.stVal to FALSE and LGOS.SimSt to FALSE (and keeps LGOS.St=FALSE); state: wait for normal GOOSE 14. DUT changes LGOS.St to TRUE (and keeps LGOS.SimSt=FALSE); state: subscription normal GOOSE | | |
| Test description  Below, Publisher 1 and Publisher 2 send same GOOSE differing only in Simulation bits. Publisher 3 sends different GOOSE messages. Publisher 1/2 are supervised by LGOS, publisher 3 is supervised by LGOS3.   1. LPHD.Sim=FALSE or not present 2. Force the DUT to ignore simulated GOOSE messages when LPHD.Sim is present 3. Publisher1 and Publisher3 sends GOOSE message with a new data value with Simulation off 4. Publisher2 sends GOOSE message with a new data value with Simulation set 5. Publisher1 stops sending GOOSE messages 6. Publisher2 stops sending GOOSE messages 7. LPHD.Sim=TRUE 8. Force the DUT to accept simulated GOOSE messages 9. Publisher1 and Publisher 3 sends GOOSE message with a new data value with Simulation off 10. Then publisher2 starts sending GOOSE message with Simulation set 11. Publisher2 sends GOOSE message with a new data value with Simulation set 12. Publisher2 stops sending GOOSE messages with Simulation set 13. Publisher1 sends GOOSE message with a new data value with Simulation off 14. Publisher1 stops sending GOOSE message with Simulation off 15. Force DUT to accept normal GOOSE messages 16. Publisher1 sends GOOSE message with a new data value with Simulation off | | |
| Comment  Note: LGOS is optional and only verified when available. When LGOS is available the LGOS.SimSt is optional | | |

|  |  |  |
| --- | --- | --- |
| **sGosN5** | **No GOOSE message** | Passed  Failed  Inconclusive |
| IEC 61850-7-2 Subclause 18.2.3  IEC 61850-8-1 Subclause 18.1, PIXIT: Gs2 | | |
| Expected result  2. DUT indicates that subscribed GOOSE1 and GOOSE2 message are received (PIXIT). GOOSE2 is always received in the next steps.   1. DUT indicates that subscribed GOOSE1 message isn’t received (PIXIT), 2. DUT indicates that subscribed GOOSE1 message is received again (PIXIT) 3. DUT indicates that subscribed GOOSE1 message isn’t received (PIXIT) 4. DUT shall process new state value(s) of GOOSE1 | | |
| Test description   1. Test engineer configures the DUT as specified to subscribe 2 different GOOSE messages: GOOSE1 from Publisher1 and GOOSE2 from Publisher2 2. Publisher1 sends correct GOOSE1 and Publisher2 sends correct GOOSE2 message with no value changes (same stNum) 3. Publisher1 is disconnected from the network, continues to send GOOSE1 messages for 30 seconds with no state change   (e.g. same stNum as step 2).   1. Publisher1 is reconnected to the network and continues to send GOOSE1 messages (same stNum) 2. Publisher1 is disconnected from the network, continues to send GOOSE1 messages for 30 seconds with no state change   (e.g. same stNum as step 2).   1. Publisher1 is reconnected to the network and continues sends GOOSE1 messages indicating a state change (incremented   stNum, sqNum other than 0) | | |
| Comment | | |