Herschel **HIFI**	**PIPE GW TEST REPORT**	**Doc. no. : SRON-U/HIFI/TN/2003-005** **Issue : Issue 1** **Date : July 3, 2003** **Category :** **Page : 1 of 9**

Title **PIPE GW test report**

Prepared by : Luc Dubbeldam Date :

Checked by : Date :

Agreed by : Date :

Authorised by : Date :

File: SRON_U_HIFI_TN_2003_005 V1.0.doc

Table of contents

1 SCOPE OF WORK

2 APPLICABLE DOCUMENTS

AD-1. SRON-U/HIFI/SP/2003-001.PIPE-GW SOW.

3 CONCLUSION

PIPE-GW has passed all tests.

The proper traffic of packets from CCSsim to IEGSE and vice versa has been demonstrated.

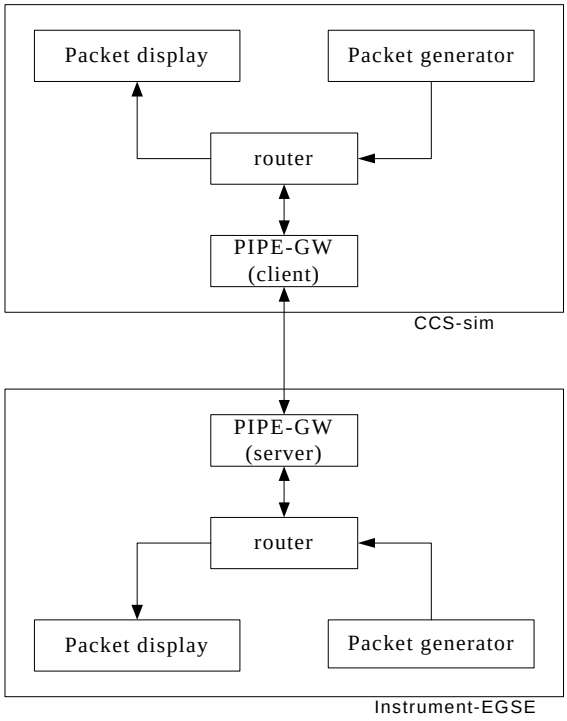
It has been demonstrated that the PIPE-GW survives and recovers from loss of environment (killed EGSE-router or killled CCSsim) without loss of outgoing packets.

It has been demonstrated that the PIPE-GW can pass 200 packets (of 924 bytes) received in a period of 200 ms in the same period of 200 ms.

During the test the content of all packets was printed to standard output. This has negative effect on the throughput and must be disabled.

4 TEST DESCRIPTION

4.1 test configuration



The figure shows the configuration. The unit under test is the PIPE-GW in the IEGSE.
In the reports the messages of both the PIPE-GW in the IEGSE and the PIPE-GW in the CCSsim will be reported when approriate.

4.2 startup

4.2.1 start PIPE-GW in nominal sequence

test steps
1. Start Router
2. Start PIPE-GW
3. Start CCS-sim

Results:
Output of PIPE-GW:

```
PIPE-GW-004 Router-connection established.
PIPE_GW-011 Request TM with APID: 1024
PIPE-GW-001 Expecting pipe-client on port:9999
PIPE-GW-002 Accepted pipe-Client
```

Output of PIPE-GW (CCS-sim)

```
PIPE-GW-004 Router-connection established.
PIPE_GW-011 Request TM with APID: 1026
PIPE-GW-012 Request TC with APID: 1024
PIPE-GW-003 Connected to: sron0203 port: 9999
```

4.2.2 start PIPE-GW in non-nominal sequence

test steps

1. Start PIPE-GW
2. Start CCS-sim
3. Start Router

Results:
Output of PIPE-GW(IEGSE):

```
nl.esa.herschel.egserouter.RouterException: java.net.ConnectException:
Connection refused
nl.esa.herschel.egserouter.RouterException: java.net.ConnectException:
Connection refused
nl.esa.herschel.egserouter.RouterException: java.net.ConnectException:
Connection refused
nl.esa.herschel.egserouter.RouterException: java.net.ConnectException:
Connection refused
nl.esa.herschel.egserouter.RouterException: java.net.ConnectException:
Connection refused
PIPE-GW-004 Router-connection established.
PIPE_GW-011 Request TM with APID: 1024
PIPE-GW-001 Expecting pipe-client on port:9999
PIPE-GW-002 Accepted pipe-Client
```

Output of PIPE-GW (CCS-sim)

```
PIPE-GW-004 Router-connection established.
PIPE_GW-011 Request TM with APID: 1026
PIPE-GW-012 Request TC with APID: 1024
PIPE-GW-003 Connected to: sron0203 port: 9999
```

4.3 Packet distribution nominal mode

4.3.1 Check alive packets

Results:
Output of PIPE-GW (IEGSE):

```
PIPE-GW-014 (0, 0) write to PIPE
```

Output of PIPE-GW (CCS-sim)

```
readPipeMessage: header
0000 |  11 00 00 18 00 00 00 00 fa de
PIPE-GW-013 (c08c000, 0) received from PIPE
readPipeMessage: packet
0000 |  0c 08 c0 00 00 0b 00 00 00 00 3f 01 75 21 00 00 00 00
```

4.3.2 TM distribution

Send a TM packet from the CCS-sim
Results:

Output of PIPE-GW (IEGSE):

```
readPipeMessage: header
0000 |  20 00 03 a2 ab cd c0 00 fa de
PIPE-GW-013 (c02c000, abcdc000) received from PIPE
readPipeMessage: packet
0000 |  0c 02 c0 00 03 95 00 03 19 00 3f 01 75 f8 00 00 00 01 00 00
                                                    (remove text to keep report readable)
0860 |  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0880 |  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0900 |  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0920 |  00 00 3d 05
PIPE-GW-014 (0, 0) write to PIPE
```

4.3.3 RC distribution + succes verification report

Send a TC packet from the CCS.
Generate a verification report-succes

Results:
Output of PIPE-GW (IEGSE):

```
readPipeMessage: header
0000 |  44 00 00 5c ab cd c0 00 fa de
PIPE-GW-013 (1c00c000, abcdc000) received from PIPE
readPipeMessage: packet
0000 |  1c 00 c0 00 00 4f 00 08 04 00 05 01 00 00 00 00 00 00 00 00
0020 |  00 00 00 01 02 03 04 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 |  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 |  00 00 00 00 00 00 00 00 00 00 01 00 01 01 01 02 02 00 02 01
0080 |  02 02 04 00 0c 0d
PIPE-GW-011 (1c00c000, abcdc000) stored
PIPE-GW-012 (1c00c000, abcdc000) verified
PIPE-GW-014 (1c00c000, abcdc000) write to PIPE

writePipeMessage: header
0000 |  50 00 00 1c ab cd c0 00 fa de
writePipeMessage: packet
0000 |  0c 00 c0 01 00 0f 00 01 01 00 3f 01 77 14 00 00 1c 00 c0 00
0020 |  5b 0a
```

4.3.4 RC distribution + failure verification report

Send a TC packet from the CCS.
Generate a verification report-failure

Results:
Output of PIPE-GW (IEGSE):

```
readPipeMessage: header
0000 |  44 00 00 5c ab cd c0 00 fa de
PIPE-GW-013 (1c00c000, abcdc000) received from PIPE
readPipeMessage: packet
0000 |  1c 00 c0 00 00 4f 00 08 04 00 05 01 00 00 00 00 00 00 00 00
0020 |  00 00 00 01 02 03 04 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 |  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
0060 |  00 00 00 00 00 00 00 00 00 00 01 00 01 01 01 02 02 00 02 01
0080 |  02 02 04 00 0c 0d
PIPE-GW-011 (1c00c000, abcdc000) stored
PIPE-GW-012 (1c00c000, abcdc000) verified
PIPE-GW-014 (1c00c000, abcdc000) write to PIPE

writePipeMessage: header
0000 |  51 00 00 1c ab cd c0 00 fa de
writePipeMessage: packet
0000 |  0c 00 c0 01 00 0f 00 01 02 00 3f 01 77 df 00 00 1c 00 c0 00
0020 |  63 af
```

4.3.5 RM distribution

Send TM packet to CCS-sim

Results:
Output of PIPE-GW (CCS-sim):

```
readPipeMessage: header
0000 |  10 00 03 a2 00 00 00 00 fa de
PIPE-GW-013 (c02c000, 0) received from PIPE
readPipeMessage: packet
0000 |  0c 02 c0 00 03 95 00 03 19 00 3f 01 7a 99 00 00 00 01 00 00
                                                    (remove text to keep report readable)
0860 |  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0880 |  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0900 |  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0920 |  00 00 21 52
```

4.4 Error recovery

4.4.1 IEGSE without router

Send packets from CCSsim with a router-failure.

Test steps
1. Start Router
2. Start PIPE-GW
3. Start CCS-sim
4. Kill the router
5. Generate 10 TC packets at CCSsim for IEGSE.
6. Restart router

Result 1: Generation of HK-packets stops at step 4 and continues at step 6.

Result 2: at PIPE-GW(IEGSE) filtering lines containing PIPE-GW)

```
nl.esa.herschel.egserouter.RouterException: java.net.ConnectException:
Connection refused
nl.esa.herschel.egserouter.RouterException: java.net.ConnectException:
Connection refused
nl.esa.herschel.egserouter.RouterException: java.net.ConnectException:
Connection refused
PIPE-GW-004 Router-connection established.
```

```
PIPE-GW-001 Expecting pipe-client on port:9999
PIPE-GW-002 Accepted pipe-Client
PIPE-GW-014 (0, 0) write to PIPE
PIPE-GW-013 (1c00c000, abcdc000) received from PIPE
PIPE-GW-011 (1c00c000, abcdc000) stored
PIPE-GW-013 (1c00c001, abcdc001) received from PIPE
PIPE-GW-011 (1c00c001, abcdc001) stored
PIPE-GW-013 (1c00c002, abcdc002) received from PIPE
PIPE-GW-011 (1c00c002, abcdc002) stored
PIPE-GW-013 (1c00c003, abcdc003) received from PIPE
PIPE-GW-011 (1c00c003, abcdc003) stored
PIPE-GW-013 (1c00c004, abcdc004) received from PIPE
PIPE-GW-011 (1c00c004, abcdc004) stored
PIPE-GW-013 (1c00c005, abcdc005) received from PIPE
PIPE-GW-011 (1c00c005, abcdc005) stored
PIPE-GW-013 (1c00c006, abcdc006) received from PIPE
PIPE-GW-011 (1c00c006, abcdc006) stored
PIPE-GW-013 (1c00c007, abcdc007) received from PIPE
PIPE-GW-011 (1c00c007, abcdc007) stored
PIPE-GW-013 (1c00c008, abcdc008) received from PIPE
PIPE-GW-011 (1c00c008, abcdc008) stored
PIPE-GW-013 (1c00c009, abcdc009) received from PIPE
PIPE-GW-011 (1c00c009, abcdc009) stored
```

4.4.2 IEGSE without CCSsim

Send packets from CCSsim with a router-failure.

Test steps
1. Start Router
2. Start PIPE-GW
3. Start CCS-sim
4. Kill the CCSsim
5. Generate 10 packets at IEGSE for CCSsim.
6. Restart CCSsim

Result: at PIPE-GW(CCSsim) filtering lines containing PIPE-GW)

```
PIPE-GW-004 Router-connection established.
PIPE-GW-012 Request TC with APID: 1024
PIPE-GW-003 Connected to: sron0203 port: 9999
PIPE-GW-013 (c08c008, 0) received from PIPE
PIPE-GW-013 (c02c000, 0) received from PIPE
PIPE-GW-013 (c02c001, 0) received from PIPE
PIPE-GW-013 (c02c002, 0) received from PIPE
PIPE-GW-013 (c02c003, 0) received from PIPE
PIPE-GW-013 (c02c004, 0) received from PIPE
PIPE-GW-013 (c02c005, 0) received from PIPE
PIPE-GW-013 (c02c006, 0) received from PIPE
PIPE-GW-013 (c02c007, 0) received from PIPE
PIPE-GW-013 (c02c008, 0) received from PIPE
PIPE-GW-013 (c02c009, 0) received from PIPE
```

4.4.3 TC without verification

Send a TC packet from the CCS.
Do not reply with verification report
Send a TC packet with the same request-ID.

Result:
Output of PIPE-GW (IEGSE)

```
PIPE-GW-013 (1c00c000, abcdc000) received from PIPE
PIPE-GW-011 (1c00c000, abcdc000) stored
PIPE-GW-013 (1c00c000, abcdc000) received from PIPE
PIPE-GW-011 (1c00c000, abcdc000) stored
PIPE-GW-103 duplicated packet-ID: 1c00c000
```

4.4.4 Unknown Type/subtype

Attempt to send a connection report (17,2) to the CCSsim
Output of PIPE-GW (IEGSE)

```
PipeWarning: PIPE-GW-101 Unknown type/subtype (17,2)
```

4.4.5 Wrong syncword

Send a message with wrong sync-word
For this test the PIPE-GW(CCSsim) was recompiled with another value for the sync-word.
Result:
In general both PIPE-GWs break the connection as soon as the other party sends a message.

Output of PIPE-GW(IEGSE)

```
PipeError: PIPE-GW-112 Error reading PIPE header
PIPE-GW-001 Expecting pipe-client on port:9999
PIPE-GW-002 Accepted pipe-Client
```

4.5 Performance test

4.5.1 Repeat RC distribution + succes verification report

Repeat the following steps

1. Send a TC packet from the CCS.
2. Generate a verification report-succes

Result:
After more than 35000 cycles:

- All TC packets were passed from CCS-sim to router
- All verification reports were sent to CCS with proper request-ID
- No unexpected errors occurred.

4.5.2 Measure throughput.

For this test the PIPE-GW of the CCSsim was adapted.
Generate 200 packets in a burst and write then without delays to the PIPE-GW (IEGSE).
The Pipe-GW was adapted to time-label every incoming packet.
Result:

The Pipe-GW is slow with incoming packets, even when the output is not printed on screen but written to a file. 200 packets generated by CCSsim within a second are received by the PIPE-GW in a period of 5 seconds.

To improve this the packet dump of the output was disabled.
Results:
PIPE-GW(CCSsim) send 200 packets of 924 bytes in 200 ms
PIPE-GW(IEGSE) receives 200 packets in 200 ms.