TECHNICAL REPORT

TR-260
DCF functional and performance Test Plan

Issue: 1
Issue Date: November 2011

Notice

The Broadband Forum is a non-profit corporation organized to create guidelines for broadband network system development and deployment. This Broadband Forum Technical Report has been approved by members of the Forum. This Broadband Forum Technical Report is not binding on the Broadband Forum, any of its members, or any developer or service provider. This Broadband Forum Technical Report is subject to change, but only with approval of members of the Forum. This Technical Report is copyrighted by the Broadband Forum, and all rights are reserved. Portions of this Technical Report may be copyrighted by Broadband Forum members.

This Broadband Forum Technical Report is provided AS IS, WITH ALL FAULTS. ANY PERSON HOLDING A COPYRIGHT IN THIS BROADBAND FORUM TECHNICAL REPORT, OR ANY PORTION THEREOF, DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY:

(A)   OF ACCURACY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE;
(B)   THAT THE CONTENTS OF THIS BROADBAND FORUM TECHNICAL REPORT ARE SUITABLE FOR ANY PURPOSE, EVEN IF THAT PURPOSE IS KNOWN TO THE COPYRIGHT HOLDER;
(C)   THAT THE IMPLEMENTATION OF THE CONTENTS OF THE TECHNICAL REPORT WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

By using this Broadband Forum Technical Report, users acknowledge that implementation may require licenses to patents. The Broadband Forum encourages but does not require its members to identify such patents. For a list of declarations made by Broadband Forum member companies, please see http://www.broadband-forum.org. No assurance is given that licenses to patents necessary to implement this Technical Report will be available for license at all or on reasonable and non-discriminatory terms.

ANY PERSON HOLDING A COPYRIGHT IN THIS BROADBAND FORUM TECHNICAL REPORT, OR ANY PORTION THEREOF, DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW (A) ANY LIABILITY (INCLUDING DIRECT, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES UNDER ANY LEGAL THEORY) ARISING FROM OR RELATED TO THE USE OF OR RELIANCE UPON THIS TECHNICAL REPORT; AND (B) ANY OBLIGATION TO UPDATE OR CORRECT THIS TECHNICAL REPORT.

Broadband Forum Technical Reports may be copied, downloaded, stored on a server or otherwise re-distributed in their entirety only, and may not be modified without the advance written permission of the Broadband Forum.

The text of this notice must be included in all copies of this Broadband Forum Technical Report.

Issue History

Issue Number	Issue Date	Issue Editor	Changes
1	November 2011	Christopher Croot, BT	Original

Comments or questions about this Broadband Forum Technical Report should be directed to info@broadband-forum.org.

Editor	Christopher Croot	BT
O&NM WG Chair	Peter Adams	ADTRAN
Vice Chair	Moti Morgenstern	ECI
Chief Editor	Michael Hanrahan	Huawei

November 2011   3 of 43

Table of Contents

List of Figures

List of Tables

Executive Summary

TR-198 [5], "*DQS: DQM systems functional architecture and requirements*" defines requirements for DQM-ready systems. More specifically, the Data Collection Function (DCF) with related interfaces is well defined in TR-198 [5] and it is of importance for operators in order to improve their DQM strategies by introducing this advanced collection capability in their networks. TR-260 specifies functional and performance tests for the DCF and is aimed to help the verification and maturing of DCF implementations.

November 2011   6 of 43

1   Purpose and Scope

1.1   Purpose

The purpose of this Technical Report is to specify functional and performance tests and appropriate test setups for verifying DCF implementations against the relevant requirements specified in TR-198 [5].

1.2   Scope

TR-260 applies to Data Collection Function implementations described in Appendix C/TR-198 [5].

The functional and performance tests of this Test Plan are mainly focused on the DCF Northbound interface primitives (i.e. the G interface in TR-198 [5]) as defined in Section 6.3.1/TR-198 [5]. Other general aspects and data attributes are also tested such as data presentation and validity. This Technical Report, like TR-198 [5], is focused on DSL technologies that are currently addressed by ITU-T G.997.1 [7].

2   References and Terminology

2.1   Conventions

In this Technical Report, several words are used to signify the requirements of the specification. These words are always capitalized. More information can be found be in RFC 2119 [6].

MUST	This word, or the term "REQUIRED", means that the definition is an absolute requirement of the specification.
MUST NOT	This phrase means that the definition is an absolute prohibition of the specification.
SHOULD	This word, or the adjective "RECOMMENDED", means that there could exist valid reasons in particular circumstances to ignore this item, but the full implications need to be understood and carefully weighed before choosing a different course.
SHOULD NOT	This phrase, or the phrase "NOT RECOMMENDED" means that there could exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications need to be understood and the case carefully weighed before implementing any behaviour described with this label.
MAY	This word, or the adjective "OPTIONAL", means that this item is one of an allowed set of alternatives. An implementation that does not include this option MUST be prepared to inter-operate with another implementation that does include the option.

2.2   References

The following references are of relevance to this Technical Report. At the time of publication, the editions indicated were valid. All references are subject to revision; users of this Technical Report are therefore encouraged to investigate the possibility of applying the most recent edition of the references listed below.

A list of currently valid Broadband Forum Technical Reports is published at www.broadband-forum.org.

Document	Title	Source	Year
[1] TR-067	*ADSL Interoperability Test Plan*	Broadband Forum	2006
[2] TR-100	*ADSL2/ADSL2plus Performance Test Plan*	Broadband Forum	2007
[3] TR-114	*VDSL2 Performance Test Plan*	Broadband Forum	2010
[4] TR-188	*DSL Quality Suite*	Broadband Forum	2010
[5] TR-198	*DQS: DQM systems functional architecture and requirements*	Broadband Forum	2010
[6] RFC 2119	*Key words for use in RFCs to Indicate Requirement Levels*	IETF	1997
[7] G.997.1	*Physical layer management for digital subscriber line (DSL) transceivers*	ITU	2009

2.3   Definitions

No terminology is used in this Technical Report.

2.4   Abbreviations

This Technical Report uses the following abbreviations:

CPE	Customer Premise Equipment
DCF	Data Collection Function
DQM	DSL Quality Management
EM	Element Manager
ME	Management Entity
MIB	Management Information Base
NE	Network Element
SUT	System Under Test
TR	Technical Report
WG	Working Group

3   Technical Report Impact

3.1   Energy Efficiency

The scope of TR-260 does not specifically cover improvements in energy efficiency, the tests described within are meant to support DQM algorithms and techniques some of which could enable improvements in energy efficiency.

3.2   IPv6

TR-260 has no impact on IPv6.

3.3   Security

The interfaces being tested in TR-260 will need to be secured against abuse using appropriate security practices.

3.4   Privacy

TR-260 has no impact on privacy.

4   DCF Testing Environment

This section specifies the testing environment for both functional and performance tests. It contains all the test configurations and settings for the System Under Test (SUT). Collection tasks are also specified.

In the functional tests, all the DCF functionalities shall be tested in a basic setup, to verify the compliance with the requirements on primitives and related behaviours.

In the performance tests the DCF shall be tested in a loaded setup, to verify the required performance capabilities of data collection are met.

4.1   Test configurations for DCF functional tests

The general test configuration for functional tests encompasses a Network Element (NE) configured with at least 2 active DSL lines connected to 2 CPE modems. The DCF is connected to a server, from now on indicated as a DCF manager, which shall be able to manage it via the DCF Northbound interface primitives specified in TR-198 [5] The Data Collection Function Northbound interface is denoted "G" in the Figure 1/TR-198 [5]. The Data Collection Function Southbound interface is denoted "A" and "B" in the Figure 1/TR-198 [5]. For the tests specified in this document the DCF manager will also be used as the server the DCF sends data reports to. The interconnection between the DCF manager and the DCF is realised via a management network which, in the lab environment, shall be as simple as possible and shall not introduce bottlenecks in the data transfer. Regardless of the DCF architecture, the interconnection between the NE and the router at the edge of the management network shall be realised via the in-band management channel in order to take into account the constraints of this type of channel. A loop simulator and a noise generator shall be connected randomly to one of the lines involved in the test to be able to cause a variation of values for the line parameters and, in turn, affect the data collected.

The setup for a Distributed DCF integrated in the NE is shown in Figure 1.
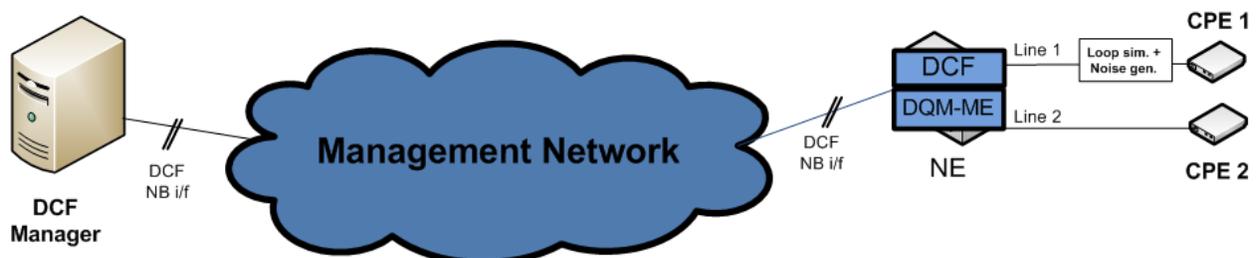
Figure 1 – Test Configuration for functional tests - Distributed DCF in NE

November 2011   12 of 43

The setup for a Distributed DCF external to the NE is shown in Figure 2 (single NE case) and Figure 3 (multiple NE case).
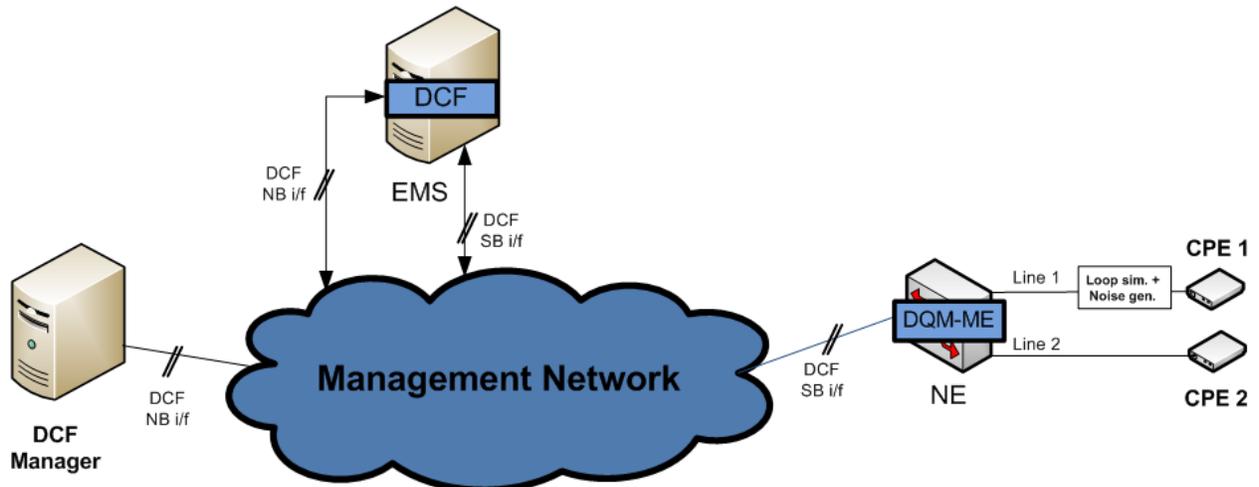
Figure 2 – Test Configuration for functional tests - Distributed DCF external, connected to a single NE
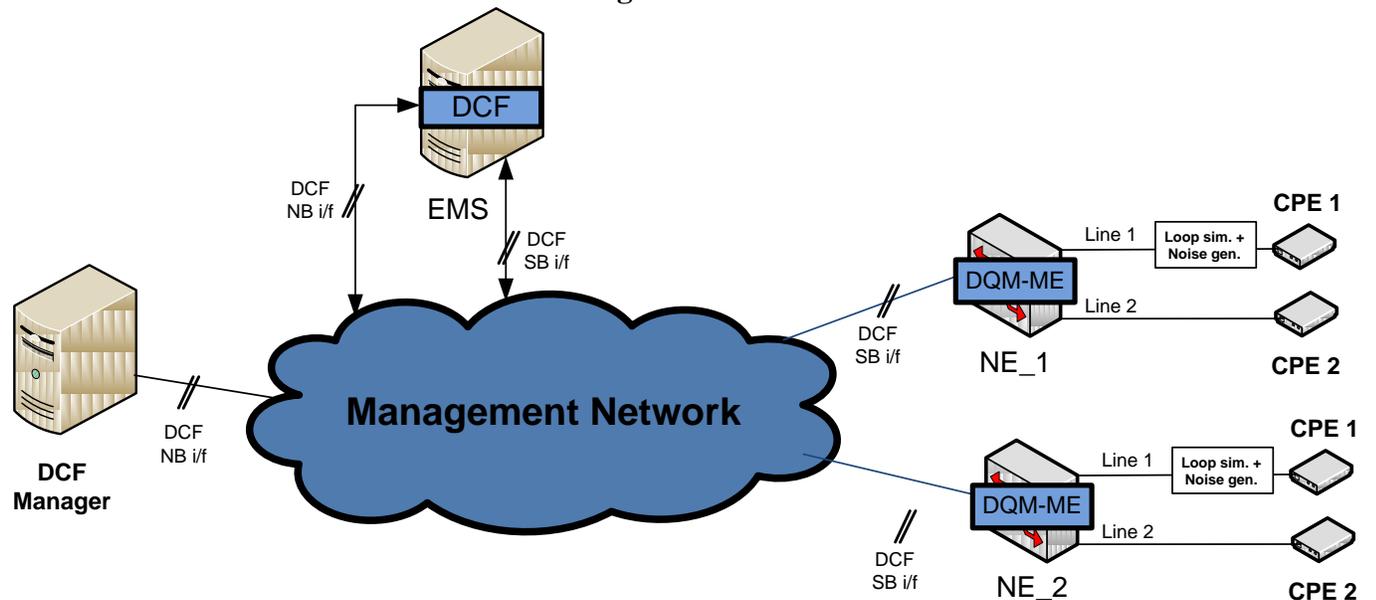
Figure 3 – Test Configuration for functional tests - Distributed DCF external, connected to multiple NEs

The setup for a Centralised DCF is for further study.

November 2011   13 of 43

4.2   Test configurations for DCF performance tests

All general considerations reported in the first paragraph of Section 4.1 apply. The main difference in the setup for performance tests in that a high number of lines have to be configured and active to create a load in the collection of data. The NE (or possibly the NEs) involved in the tests MUST be fully equipped.

To practically achieve this configuration without the need of a huge number of CPE modems, the lines MUST be configured to stay in L0 mode even if the modem is disconnected[1].

The following procedure can be used to activate a high number of NE lines:

1. a block of lines (e.g. all the lines of a linecard) is configured and the connected modems activated;
2. after the handshake, a showtime freeze command is sent to these lines;
3. the modems are disconnected;
4. repeat the steps above until all the NE lines are active;
5. for the last block, the modems can be left connected.

It may well be that the same effect of a high number of lines loading the DCF with data can be obtained in two other ways:

a) "freezing", via a special command, the MIB content of the lines once they have reached showtime;
b) provisioning the lines as logical entities hence creating the associated instance in the NE's MIB tree with all parameters collectable, regardless of their actual value. In latter case the lines would not even need to be synchronized but then the Collection Tasks specified in Section 4.4.3 shall be configured without any "collection filter".

Nevertheless the actual ability to load the data collection via options a) or b) above needs to be proved upfront.

The figures below show the test configuration assuming a limited number of modems are connected to the NE, while all the other lines are kept in L0 state.

The setup for a Distributed DCF integrated in the NE is shown in Figure 4.

---

[1] *This setting is often referred to as "showtime freeze" or "always connected".*

November 2011   14 of 43
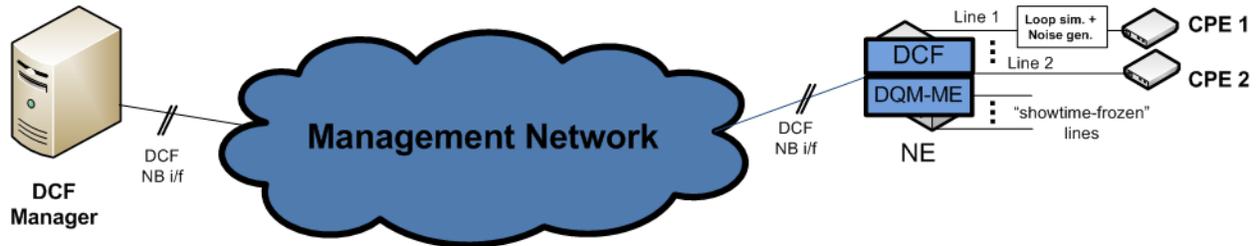
Figure 4 – Test Configuration for performance tests - Distributed DCF in the NE

The setup for a Distributed DCF external to the NE is shown in Figure 5.
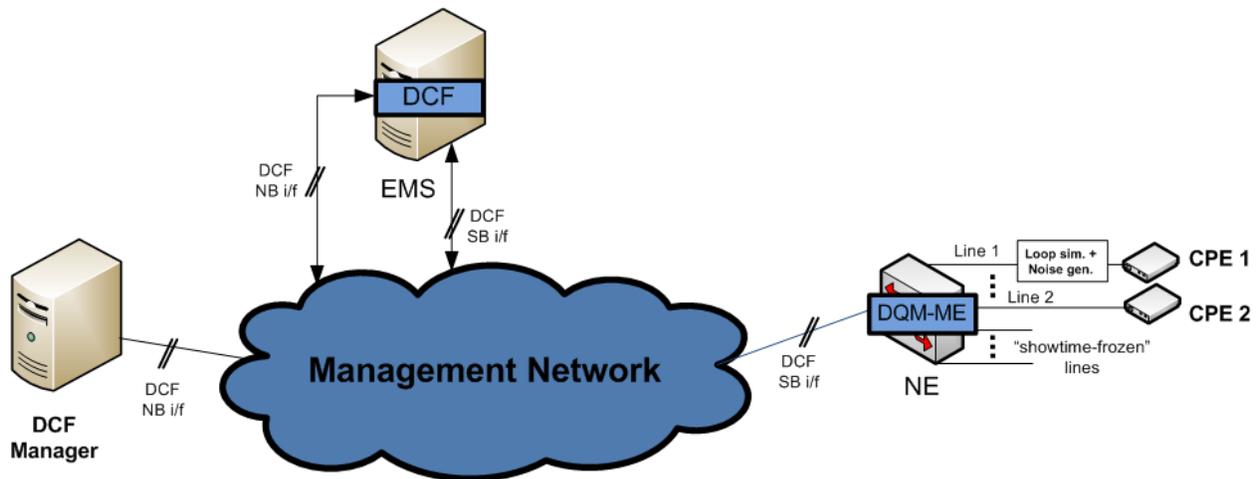
Figure 5 – Test Configuration for performance tests - Distributed DCF external to NE

Figure 6 – Test Configuration for performance tests - Distributed DCF external to multiple NEs

The setup for a Centralised DCF is for further study.

4.3   Configurations of DSL lines

The configuration of the DSL lines is not specifically relevant for the execution of the tests specified in this Test Plan. Furthermore these tests are applicable to any DSL flavour (e.g. ADSL, ADSL2plus VDSL2,) as such the number of parameters and collectable data associated to different DSL modes of operation varies.

In the remainder of this document it is assumed that:

   a) Line 1 belongs to a VDSL2 linecard and is configured with a profile-line combination as defined in TR-114 [3];
   b) Line 2 belongs to an ADSL2/2pluslinecard and is configured with a specific test profile as defined in TR-100 [2].

For the Push_upload test (see Section 5.2.2) and Data validity test (see Section 5.2.6), correct data collection is verified twice, first when Line 2 belongs to an ADSL2/2plus linecard and configured as above and secondly when:

   c) Line 2 belongs to an ADSL linecard and it is configured as per one of the settings defined in TR-067 [1].

4.4   Definition of Collection Tasks
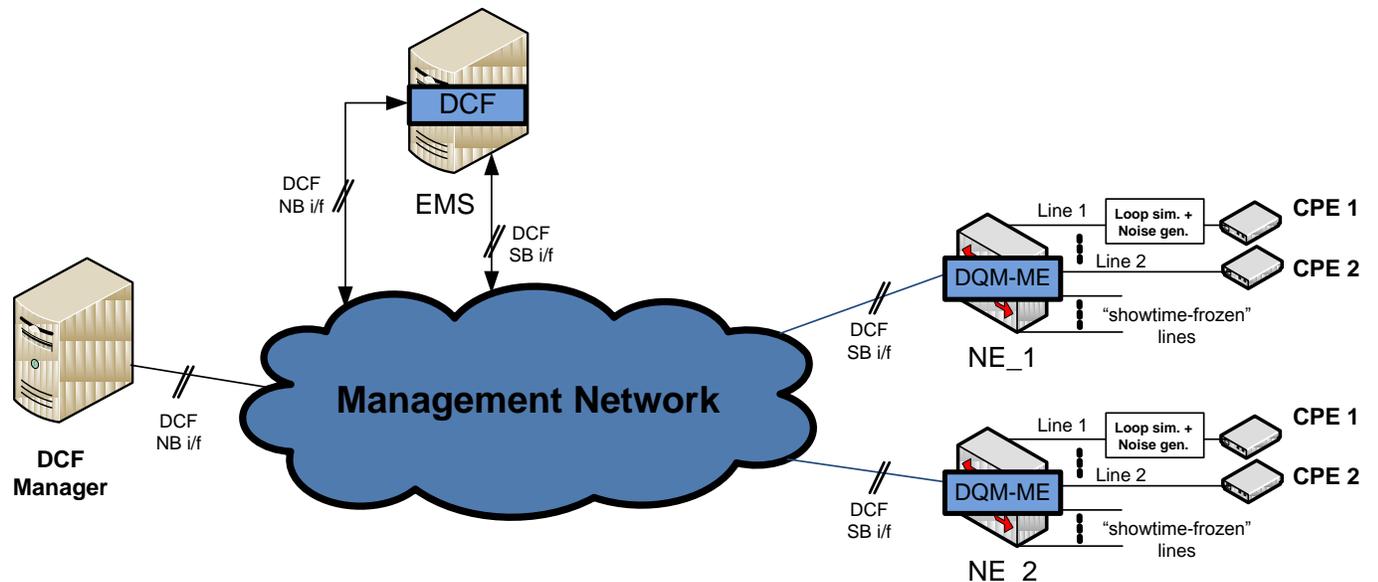
This section specifies the configurations of different CollectionTasks on the DCF as deemed necessary to perform the Test Cases specified in this Test Plan.

The Task Information fields are those in Table 10/TR-198 [5] for the Create_collection_task.request primitive. Each Task Information field is specified in Table 11/TR-198 [5].

For sake of simplicity, lines are indicated by a single number. In actual implementation an unambiguous notation shall refer to a line in the context of a network hierarchy: {NE_ID, Line_ID} where Line_ID is, for example, defined by rack, shelf, slot, port number.

4.4.1   Collection Tasks for functional tests

Collection Task 2

This collection task, once activated, collects scalar and framing parameters from Line 2 every 15 minutes for 6 hours and then stops. Data is automatically uploaded to the DCF manager every 30 minutes. Data is not collected from a line that is in a "defect" status. Defect is defined in Section 3.5/G.997.1 [7].

Task Information Fields	Value
task_name	CollectionTask2
resource_list {parameter_list,collection_frequency,upload_frequency}	Line: 2 [scalar parameters,15min,30min] [framing parameters,15min,30min]
collection_filter	Line: 2 (xdsl2LineStatusXtuc = 'noDefect')
task_duration	6h
priority_request_flag (if supported)	-

Collection Task 3

This collection task, once activated, collects performance, vectorial, status and xTU information parameters from Lines 1 and 2 every 15 minutes for 24 hour and then stops. All data groups are automatically uploaded to the DCF manager every 15 minutes with the exception of vectorial parameters which are uploaded every hour. Data is not collected from a line that is in a "defect" status.

Task Information Fields	Value
task_name	CollectionTask3
resource_list {parameter_list,collection_frequency,upload_frequency}	Line: 1, 2 [performance monitoring parameters,15min,15min] [vectorial parameters,15min,1h] [status parameters,15min,15m] [xTU information parameters,15min,15min]
collection_filter	Line: 1, 2 (xdsl2LineStatusXtuc = 'noDefect')
task_duration	24h
priority_request_flag (if supported)	-

Collection Task Multi_NE

This collection task, once activated, collects performance, vectorial, status and xTU information parameters from Lines 1 and 2 of Network Elements NE_1 and NE_2 (see Figure 6) every 15

minutes for 24 hours and then stops. Data is automatically uploaded to the DCF manager every 15 minutes. Data is not collected from a line that is in a "defect" status.

Task Information Fields	Value
task_name	CollectionTask_Multi_NE
resource_list {parameter_list,collection_frequency,upload_frequency}	NE: NE_1, NE_2 {Line: 1, 2 [performance monitoring parameters,15min,15min] [vectorial parameters,15min,1h] [status parameters,15min,15min] [xTU information parameters,15min,15min] }
collection_filter	Line: 1, 2 (xdsl2LineStatusXtuc = 'noDefect')
task_duration	24h
Priority_request_flag (if supported)	-

4.4.2   Collection Tasks for Maximum load tests

These tasks and the tests specified in Section 7 are useful to do an investigation of the maximum DCF load if the performance tests specified in Section 6 are not met. These tests may be especially useful where the DCF is not integrated in the NE and data gathered via SNMP which may introduce bottlenecks at different levels (e.g. max parameters per line, max lines per linecard or max lines per NE).

This investigation is beyond the actual verification of TR-198 [5] compliance and it is included only as an aid to correctly test the DCF implementation.

CollectionTask-MaxParams

This collection task is useful to verify the capability of a DCF to collect all parameters required in TR-198 [5] on a single xDSL line on the NE. The collection frequency (and associated period T_{coll}) depends on the parameter group and the duration is 1 day. Data is automatically uploaded to the DCF manager with an upload frequency (and associated period T_{upload}) identical to the collection frequency. The line under test MUST not be on the "defect" status.

Task Information Fields	Value
task_name	Max_Params_Single_Line
resource_list {parameter_list,collection_frequency,upload_frequency}	Line: 1 line on the NE (e.g. line 1 on the board 1)
	[scalar parameters, 15min, 15min]
	[framing parameters, 24h, 24h]
	[performance monitoring parameters, 15min,15min]
	[vectorial parameters, 15min, 15min]
	[status parameters, 15min,15min]
	[xTU information parameters, 24h, 24h]
	[Vector of Profiles parameters, 15min,15min]
	[Threshold crossing notifications, 15min,15min]
collection_filter	Line: 1 line on the NE (e.g. line 1 on the board 1)
	(xdsl2LineStatusXtuc = 'noDefect')
task_duration	24h
priority_request_flag (if supported)	-

CollectionTask-MaxLinesPerLinecard

This collection task is useful to verify the capability of a DCF to collect all parameters required in TR-198 [5] over the maximum number of xDSL lines on a single line card as declared by the manufacturer. This test stresses the board performance to verify the ability of the on-board hardware/software to manage the data collection load. The collection frequency (and associated period T_{coll}) depends on the parameter group and the duration is 1 day. Data is automatically uploaded to the DCF manager with an upload frequency (and associated period T_{upload}) identical to the collection frequency. The lines under test MUST not be on the "defect" status.

Task Information Fields	Value
task_name	Max_Lines_Per_Linecard

resource_list {parameter_list,collection_frequency,upload_frequency}	Line: the declared maximum number (k) of lines of a line card on the NE (e.g. lines 1 to k on the board 1) [scalar parameters, 15min, 15min] [framing parameters, 24h, 24h] [performance monitoring parameters, 15min,15min] [vectorial parameters, 15min, 15min] [status parameters, 15min,15min] [xTU information parameters, 24h, 24h] [Vector of Profiles parameters, 15min,15min] [Threshold crossing notifications, 15min,15min]
collection_filter	Line: the declared maximum number (k) of lines of a line card on the NE (e.g. lines 1 to k on the board 1) (xdsl2LineStatusXtuc = 'noDefect')
task_duration	24h
priority_request_flag (if supported)	-

CollectionTask-MaxLinesPerNE

This collection task is useful to verify the capability of a DCF to collect all parameters required in TR-198 [5] over the maximum number of xDSL lines on the NE as declared by the manufacturer. The collection frequency (and associated period T_{coll}) depends on the parameter group and the duration is 1 day. Data is automatically uploaded to the DCF manager with an upload frequency (and associated period T_{upload}) identical to the collection frequency. Data is not collected from a line that is in a "defect" status.

Task Information Fields	Value
task_name	Max_Lines_Per_NE
resource_list {parameter_list,collection_frequency,upload_frequency}	Line: the declared maximum number (m) of lines on the NE (e.g. lines 1 to k on n boards so that k*n=m)

	[scalar parameters, 15min, 15min]
	[framing parameters, 24h, 24h]
	[performance monitoring parameters, 15min,15min]
	[vectorial parameters, 15min, 15min]
	[status parameters, 15min,15min]
	[xTU information parameters, 24h, 24h]
	[Vector of Profiles parameters, 15min,15min]
	[Threshold crossing notifications, 15min,15min]
collection_filter	Line: the declared maximum number (m) of lines on the NE (e.g. lines 1 to k on n boards so that k*n=m) (xdsl2LineStatusXtuc = 'noDefect')
task_duration	24h
priority_request_flag (if supported)	-

4.4.3   Collection Tasks for performance tests

Collection Task Ref_Perf_BckGnd_Task_TR-198

This collection task is taken from Table 13/TR-198 [5] and represents the multi-line background collection activities of the combined collection task defined in TR-198 [5] as a reference for the DCF performance requirements. Once activated, this collection task collects all parameters from all the active lines of the NE. The collection frequency (and associated period T_{coll}) depends on the parameter group and the duration is 3 days. Data is automatically uploaded to the DCF manager with an upload frequency (and associated period T_{upload}) identical to the collection frequency. Data is not collected from a line that is in a "defect" status.

Note: DCF performance requirements in TR-198 [5] for the reference combined collection task in Table 13, allow $T_{upload} = [1 \text{ to } 4] * T_{coll}$. In this Test Plan, for the performance tests: $T_{upload} = T_{coll}$.

Task Information Fields	Value
task_name	Ref_Perf_BckGnd_Task_TR-198
resource_list {parameter_list,collection_frequency,upload_frequency}	Line: all the lines of the NE [scalar parameters,15min, 15min] [framing parameters, 24h, 24h] [performance monitoring parameters,15min,15min] [vectorial parameters,6h, 6h] [status parameters,15min,15min] [xTU information parameters, 24h, 24h] [Vector of Profiles parameters,15min,15min] [Threshold crossing notifications,15min,15min]
collection_filter	Line: all the lines of the NE (xdsl2LineStatusXtuc = 'noDefect')
task_duration	72h
priority_request_flag (if supported)	-

Collection Task Ref_Perf_Task-k_TR-198

This collection task is taken from Table 13/TR-198 [5] and represents the collection activities on a single line of the combined collection task defined in TR-198 [5] as a reference for the DCF performance requirements. Once activated, this collection task collects all parameters from line "k". The collection frequency (and associated period T_{coll}) depends on the parameter group and the duration is 24 hours. Data is automatically uploaded to the DCF manager with an upload frequency (and associated period T_{upload}) identical to the collection frequency. Data is not collected from a line that is in a "defect" status. Assignment of the actual value of "k" is done during the related test.

Note: TR-198 [5] allows, for the reference combined collection task in Table 13/TR-198 [5], that $T_{upload} = [1 \text{ to } 4] * T_{coll}$.

Task Information Fields	Value
task_name	Ref_Perf_Task-k_TR-198
resource_list {parameter_list,collection_frequency,upload_frequency}	Line: k (this is a dummy value) [scalar parameters,15min, 15min] [framing parameters, 24h, 24h] [performance monitoring parameters,15min,15min] [vectorial parameters, 15min,15min] [status parameters,15min,15min] [xTU information parameters, 24h, 24h] [Vector of Profiles parameters,15min,15min] [Threshold crossing notifications,15min,15min]
collection_filter	Line: k (this is a dummy value) (xdsl2LineStatusXtuc = 'noDefect')
task_duration	24h
priority_request_flag (if supported)	-

5   DCF functional tests

This section specifies Test Cases related to the verification of the availability and correct implementation of DCF functionalities, namely Task configuration and reporting and Data Collection. The syntax for the list of parameters to be collected is based on the parameter groups and names reported in Section 6.1/TR-198 [5].

The order of the Test Cases below is such that the verification of functional aspects relies on the SUT configuration and status as left by the execution of preceding Test Cases (e.g. Task creation and activation tests first, then data upload ones and then Task deletion verifications).

This way of following the Test Case numbering allows a series of functional tests to be performed in the most efficient way. On the other hand for stand-alone execution of one specific Test Case the test engineer may need, as a propaedeutic activity, to perform the configuration steps described in the Test Procedure of some of the preceding Test Cases.

5.1   Task Configuration and Management tests

These tests aim to verify the configurability and correct functional behaviour of the DCF via the G interface as per the requirements specified in Section 6.3/TR-198 [5].

5.1.1   Create_collection_task test

Covered TR-198 [5] requirements: R-22, R-23, R-24, R-54 to 58

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Create Collection Task 2 on the DCF via the DCF manager. 2. Create Collection Task 3 on the DCF via the DCF manager.
Expected Result	1. The configurable Task Information Fields MUST be as those specified in Table 10/TR-198 [5] for the *Create_collection_task.request* primitive. 2. The syntax and semantic of Information field MUST be as specified in Table 11/ TR-198 [5] and R-54 to 58 of TR-198 [5], especially as far as mandatorily supported values are concerned. 3. The Collection Tasks MUST be created on the DCF and a *Create_collection_task.confirm* message MUST be relayed on the DCF manager with following information: 　▪ task_ID 　▪ task_status; the value of this parameter MUST be *inactive* 　▪ errorcode Note: reporting of the errorcode is OPTIONAL in this case as no error is expected.

The following test is applicable only to a DCF external, connected to multiple NEs.

Covered TR-198 [5] requirements: R-22, R-23, R-24, R-54 to 58

Test Configuration	Refer to Figure 3 Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Create Collection Task Multi_NE on the DCF via the DCF manager.
Expected Result	1. The configurable Task Information Fields MUST be as those specified in Table 10/TR-198 [5] for the *Create_collection_task.request* primitive. 2. The syntax and semantic of Information field MUST be as specified in Table 11/TR-198 [5] and R-54 to 58 of TR-198 [5], especially as far as mandatorily supported values are concerned. 3. Collection Task Multi_NE MUST be created on the DCF and a *Create_collection_task.confirm* message MUST be relayed on the DCF manager with following information: 　▪ task_ID 　▪ task_status; the value of this parameter MUST be inactive 　▪ errorcode Note: the report of the errorcode is OPTIONAL in this case as no error is expected.

November 2011 25 of 43

5.1.2 Modify CollectionTask test [Optional]

This test is mainly focused on a functionality not defined in TR-198 [5], namely the DCF ability to modify a Collection Task. The compliance to this test is OPTIONAL for the SUT.

Covered TR-198 [5] requirements: R-22, R-23, R-24, R-54 to 58.

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively. Collection Task 2 MUST be "active", while Collection Task 3 MUST be "inactive"
Test Procedure	1. Modify Collection Task 2 on the DCF via the DCF manager, changing the upload frequency of the vectorial parameters and the task duration as in Table 1. 2. Modify Collection Task 3 on the DCF via the DCF manager, changing the upload frequency of the vectorial parameters and the task duration as in Table 2. 3. Verify the effect of the two modification requests through two Retrieve_collection_task_info.request, one for each task.
Expected Result	1. The configurable Task Information Fields MUST be as those of the *Modify_collection_task.request* primitive. 2. The syntax and semantic of Information field MUST be as specified in Table 11/ TR-198 [5] and R-54 to 58 of TR-198 [5], especially as far as mandatorily supported values are concerned. 3. The Collection Tasks 2 MUST NOT be modified on the DCF and a *Modify_collection_task.confirm* message MUST be relayed on the DCF manager with following information: ▪ task_ID ▪ errorcode; the value of this parameter MUST specify "*not executed because active*" 4. The Collection Tasks 3 MUST be modified on the DCF and a *Modify_collection_task.confirm* message MUST be relayed on the DCF manager with following information: ▪ task_ID ▪ errorcode; the value of this parameter MUST be "*success*"

Table 1 - Changing parameters of Collection Task 2

Task Information Fields	Value
resource_list {parameter_list,collection_frequency,upload_frequency}	Line: 2 [scalar parameters,15min,15min] [framing parameters,15min,15min]
collection_filter	Line: 2 (xdsl2LineStatusXtuc = 'noDefect')
task_duration	24h
priority_request_flag (if supported)	-

Table 2 - Changing parameters of Collection Task 3

Task Information Fields	Value
resource_list {parameter_list,collection_frequency,upload_frequency}	[vectorial parameters,15min,15min]
collection_filter	Line: 1, 2 (xdsl2LineStatusXtuc = 'noDefect')
task_duration	2h
priority_request_flag (if supported)	-

5.1.3   Retrieve_collection_tasks test

Covered TR-198 [5] requirements: R-22, R-23, R-38.

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Request via the DCF manager the information associated to Collection Task 2. 2. Request via the DCF manager the information associated to Collection Task 3.

November 2011   27 of 43

Expected Result	1. For both Collection Task 2 and Collection Task 3, the following information MUST be presented:
	▪ task_ID
	▪ task_status
	▪ task_name
	▪ resource_list {parameter_list,collection_frequency, upload_frequency}
	▪ collection_filter
	▪ task_duration
	▪ priority_request_flag
	▪ errorcode
	Note: the report of the errorcode is OPTIONAL in this case as no error is expected.

5.1.4 Retrieve_all_collection_tasks test

Covered TR-198 [5] requirements: R-22, R-23, R-39

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not.
	Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Request via the DCF manager the information associated to all collection tasks.
Expected Result	1. For all collection tasks, the following information MUST be presented:
	▪ task_ID
	▪ task_status
	▪ errorcode
	Note 1: as an alternative to the task_ID, the DCF manager MAY show the task_name.
	Note 2: the report of the errorcode is OPTIONAL in this case as no error is expected.

5.1.5 Retrieve_all_active_collection_tasks

Covered TR-198 [5] requirements: R-22, R-23, R-40

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not.
	Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.

Test Procedure	1. Request via the DCF manager the information associated to all active collection tasks.
Expected Result	1. For all active collection tasks, the following information MUST be presented: ▪ task_ID ▪ errorcode Note 1: as an alternative to the task_ID, the DCF manager MAY show the task_name. Note 2: the report of the errorcode is OPTIONAL in this case as no error is expected.

5.2   Data Collection Tests

These tests aim to verify the correct upload of the collected data from the DCF to the DCF manager.

5.2.1   Start_collection_task test

Covered TR-198 [5] requirements: R-22, R-23, R-25 to 26, R-28 to 29

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Request via the DCF manager the start of the inactive Collection Tasks 2 and 3. 2. Request via the DCF manager the information associated to all collection tasks.
Expected Result	1. Collection Task 2 and 3 will start data collection as configured and a *Start_collection_task.confirm* message MUST be relayed on the DCF manager with following information: ▪ task_ID ▪ errorcode Note: the report of the errorcode is OPTIONAL in this case as no error is expected. 2. As a result of step 2 Collection Tasks 2 and 3 MUST appear in the retrieved list of tasks with status equal to *active*.

The following test is applicable only to a DCF external, connected to multiple NEs.

Covered TR-198 [5] requirements: R-22, R-23, R-25 to 27

Test Configuration	Refer to Figure 3 Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Request via the DCF manager the start of the inactive Collection Task Multi_NE.

Expected Result	1. Collection Task 2 and 3 are successfully started independently
	
	2. Collection Task Multi_NE will start data collection as configured and a *Start_collection_task.confirm* message MUST be relayed on the DCF manager with following information:
	
	    ▪ task_ID
	    ▪ errorcode
	
	Note: the report of the errorcode is OPTIONAL in this case as no error is expected.
	
	3. As a result of step 2 Collection Task Multi_NE MUST appear in the retrieved list of tasks with status equal to *active*.

5.2.2   Push_upload test

Covered TR-198 [5] requirements: R-22, R-23, R-30 to 33, R-44, R-47 to 49, R-53 to 59

November 2011   30 of 43

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Wait at least one hour from the start of Collection Tasks 2 and 3, then request the DCF manager to show all the data reports, associated to Collection Tasks 2 and 3, already uploaded by the DCF 2. Verify on the DCF <u>that the</u> data reports are deleted after their upload, by checking in the system folder used for their storage or other system resources appropriate for such verification 3. Create a collection task named Collection Task 3_ADSL with the same parameters as Collection Task 3, but with Line 2 belonging to an ADSL linecard and configured as per Section 4.3, point c). Furthermore only parameters defined for ADSL MUST be included in the parameters list. 4. Repeat steps 1 and 2 above. 5. Return to the original Collection Task 3 configuration, with Line 2 belonging to an ADSL2/2plus linecard.

Expected Result	1. After each upload period and for each of the above tasks a *report_upload.indicate* message MUST be relayed on the DCF manager with following information:
	  ▪ task_ID
	  ▪ report
	  ▪ errorcode
	 Note: the report of the errorcode is OPTIONAL in this case as no error is expected.
	
	 2. All the already uploaded data reports MUST be readable from the DCF manager GUI and exportable in a machine readable file format
	
	 3. In either case the content of data reports MUST conform to R-30 to 32, R-44, and R-53 to 59 of TR-198 [5]
	
	 4. The first collected data report MUST conform to R-33 of TR-198 [5]
	
	 5. As related to the check in step 2, the DCF may optionally delete the uploaded data reports as per R-49 of TR-198 [5]

5.2.3 Pull-upload test

Covered TR-198 [5] requirements: R-22, R-23, R-30 to 32, R-44 to 46, R-53 to 59

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Wait at least one hour from the start of Collection Tasks 2 and 3, then issue a *report_upload.request* (pull-upload) via the DCF manager the upload of collected data, associated to Collection Tasks 2 and 3. 2. Verify on the DCF <u>if</u> data reports are deleted after their upload, by checking in the system folder used for their storage or other system resources appropriate for such verification

Expected Result	1. For each of the above tasks a *Report_upload.confirm* message MUST be uploaded by the DCF to the DCF manager with following information:
	   ▪ task_ID
	   ▪ report
	   ▪ errorcode
	Note: the report of the errorcode is OPTIONAL in this case as no error is expected.
	
	2. The data reports MUST be readable from the DCF manager GUI and exportable in a machine readable file format
	
	3. In either case the content of data reports MUST conform to R-30 to 32, R-44, and R-53 to 59 of TR-198 [5].
	
	4. As per the definition of pull-upload, the data reports MUST only contain the data pertaining to the period between the end of last upload period and the time when the *report_upload.request* was issued.
	
	5. [Optional] As related to the check in step 2, the DCF <u>MUST NOT</u> delete the uploaded data reports. Note: this behaviour is OPTIONAL as the related requirement is not defined in TR-198 [5].

5.2.4   Collection_filter test

Covered TR-198 [5] requirements: R-22, R-23

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Wait at least one hour from the start of Collection Task 3. 2. Immediately before the current upload period expires disconnect Line 2. 3. Wait until next upload period expires then request the DCF manager to show the data reports, associated to Collection Task 3 and pertaining to that upload period.
Expected Result	1. The data reports MUST contain the parameters collected from Line 1 and MUST not contain any record or show an empty record related to Line 2.

5.2.5   Report_delete test

Covered TR-198 [5] requirements: R-22, R-23, R-50 to 52

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not.
	Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. When the clock time is hh:35, request via the DCF manager that the data reports associated to Collection Task 3 be deleted.
	2. Verify on the DCF if data reports are deleted, by checking in the system folder used for their storage or other system resources appropriate for such verification.
	3. Request via the DCF manager the information associated to Collection Task 3.
	4. Wait until the current upload period expires (i.e. 1h for vectorial parameters and 15min for all other parameter groups) plus the time needed for the upload, then verify which data reports have been uploaded by the DCF
Expected Result	1. A *Report_delete.confirm* message MUST be relayed on the DCF manager with following information:
	▪ task_ID
	▪ errorcode
	Note: the report of the errorcode is OPTIONAL in this case as no error is expected.
	2. The data reports associated to Collection Task 3 MUST be deleted on the DCF, regardless of them having been already uploaded or not.
	3. Collection Task 3 status MUST still be *active*
	4. Among the data reports uploaded after the current upload period, only those of vectorial parameters MUST be missing for the first and second quarter of the last 1h upload period.
	Data related to all the other parameters groups MUST be available for all periods, with the exception for the third quarter of the last 1h upload period for which the report may also be absent or incomplete.

5.2.6   Data validity test

This test aims to verify the correctness of the data collection comparing the data collected by DCF and the data collected directly on the NE by other ways such as command line or EM interfaces.

Covered TR-198 [5] requirements: R-44, R-59

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Via Element Manager (or CLI) GUI, retrieve Line 1 and 2 parameters that are configured for collection in Collection Tasks 2 and 3, at the end of the collection period. 2. Wait until current upload period expires then request the DCF manager to show the data reports, associated to Collection Tasks 2 and 3 and pertaining to that upload period. 3. Create a collection task named Collection Task 3_ADSL with the same parameters as Collection Task 3, but with Line 2 belonging to an ADSL linecard and configured as per Section 4.3, point c). Furthermore only parameters defined for ADSL MUST be included in the parameters list. 4. Repeat steps 1 and 2 above. 5.
Expected Result	1. Data collected via Collection Tasks 2 and 3 and those gathered via the EM (or CLI) GUI and corresponding to the same parameter and period MUST have: ▪ the same value for those parameters static/quasi-static in nature (e.g. bitrate, INP) ▪ resembling values for those parameters subject to updates during showtime (e.g. NM, Hlog)

5.2.7   Stop_collection_task test

Covered TR-198 [5] requirements: R-22, R-23, R-35 to 36

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Request the DCF, via the DCF manager, to stop the active Collection Tasks 2 and 3. 2. Request via the DCF manager the information associated to all collection tasks.
Expected Result	1. Collection Task 2 and 3 will stop data collection without deleting collected data. 2. A *Stop_collection_task.confirm* message MUST be relayed on the DCF manager with following information: ▪ task_ID ▪ errorcode Note: the report of the errorcode is OPTIONAL in this case as no error is expected. 3. As a result of step 2 Collection Task 3 MUST appear in the retrieved list of tasks with status equal to *inactive*.

The following test is applicable only to a DCF external, connected to multiple NEs.
Covered TR-198 [5] requirements: R-22, R-23, R-35 to 37

Test Configuration	Refer to Figure 3 Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Request the DCF, via the DCF manager, to stop the active Collection Task Multi_NE. 2. Request via the DCF manager the information associated to all collection tasks.
Expected Result	1. Collection Task Multi_NE will stop data collection without deleting collected data. 2. A *Stop_collection_task.confirm* message MUST be relayed on the DCF manager with following information: ▪ task_ID ▪ errorcode Note: the report of the errorcode is OPTIONAL in this case as no error is expected. 3. As a result of step 2 Collection Task Multi_NE MUST appear in the retrieved list of tasks with status equal to *inactive*.

5.2.8   Task duration test [Optional]

This test is mainly focused on a behaviour not defined in TR-198 [5], namely the DCF management of the status of expired Collection Tasks. The compliance to this test is OPTIONAL for the SUT.

Covered TR-198 [5] requirements: R-22, R-23

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Wait until the duration periods of Collection Task 2 expire. 2. Request via the DCF manager the information associated to all collection tasks.
Expected Result	1. In the collection task list Collection Tasks 2 MUST appear in the list of tasks as *completed*.

5.2.9   Delete_collection_task test

Covered TR-198 [5] requirements: R-22, R-23, R-41 to 43

Test Configuration	Refer to Figure 1 if the DCF is integrated in the NE or Figure 2 if not. Line 1 and 2 configured as per Section 4.3, points a) and b) respectively.
Test Procedure	1. Delete on the DCF, via the DCF manager, Collection Task 3. 2. Request via the DCF manager the information associated to all collection tasks. 3. Request the DCF manager to show all the data reports, associated to Collection Task 2, already uploaded by the DCF
Expected Result	1. Collection Task 3 MUST be deleted and a *Delete_collection_task.confirm* message MUST be relayed on the DCF manager with following information: ▪ task_ID ▪ errorcode Note: the report of the errorcode is OPTIONAL in this case as no error is expected. 2. As a result of step 2 Collection Task 3 MUST not appear in the retrieved list of tasks. 3. As a result of step 3 the data reports uploaded before the deletion of Collection Task 3 MUST still be available.

November 2011   37 of 43

6   DCF performance tests

These tests aim to verify that the DCF fulfils the performance requirements specified in TR-198 [5]. This means to verify that the DCF's aggregate processing/memory/data-transfer capabilities are supporting the minimum conventional load associated to a reference combined collection task as specified in Section 6.5/TR-198 [5].

November 2011   38 of 43

6.1   Performance tests under reference combined use case

Covered TR-198 [5] requirements: R-63

Test Configuration	Refer to Figure 4 if the DCF is integrated in the NE or Figure 5 if not.
	Lines belonging to VDSL2 linecards configured as per Section 4.3, point a).
	Lines belonging to ADSL2/2plus linecards as per Section 4.3, point b).
	As explained in Section 4.2, to avoid the use of huge number of CPEs, the lines MUST be configured in way that is proved to generate the same data collection load as if the line itself was terminated with a CPE and in showtime.
	Section 4.2 suggests three possible ways for doing that: ▪ putting the line in "showtime freeze" (aka "continuously sending") ▪ "freezing" the MIB content of the line once it has reached showtime ▪ creating the associated instance in the NE's MIB tree with all parameters collectable; in this case the Collection Tasks MUST have the "collection filter" off.
Test Procedure	1. Create Collection Task Ref_Perf_BckGnd_Task_TR-198 on the DCF via the DCF manager.
	2. Let N be the number representing 10% of the total number of the lines supported by the NE.
	3. Create N tasks of the type of Collection Task Ref_Perf_Task-k_TR-198 on the DCF via the DCF manager.
	4. Activate Collection Task Ref_Perf_BckGnd_Task_TR-198 and the N tasks created in step 2.
	5. Wait until a number of collection and upload periods (as configured for the specific parameter groups) pass, then request the DCF manager to show all the data reports, associated to all the collection tasks activated in step 4.
	6. Wait until the duration periods of all the collection tasks activated in step 4 expire, then request the DCF manager to show all the data reports, associated to all of those tasks.
Expected Result	1. The DCF MUST be able to collect and upload all the requested parameters for each configured collection period throughout the duration of each of the involved tasks.
	2. All the data reports generated by the configured collection/upload scheduling MUST be available on the DCF manager and contain all the parameters configured for collection.

The following test is applicable only to a DCF external, connected to multiple NEs.
Covered TR-198 [5] requirements: R-22, R-23, R-63

Test Configuration	Refer to Figure 6. Lines belonging to VDSL2 linecards configured as per Section 4.3, point a). Lines belonging to ADSL2/2plus linecards as per Section 4.3, point b). As explained in Section 4.2, to avoid the use of huge number of CPEs, the lines MUST be configured in way that is proved to generate the same data collection load as if the line itself was terminated with a CPE and in showtime. Section 4.2 suggests three possible ways for doing that: ▪ putting the line in "showtime freeze" (aka "continuously sending") ▪ "freezing" the MIB content of the line once it has reached showtime ▪ creating the associated instance in the NE's MIB tree with all parameters collectable; in this case the Collection Tasks MUST have the "collection filter" off.
Test Procedure	1. Create Collection Task Ref_Perf_BckGnd_Task_TR-198_Multi_NE on the DCF via the DCF manager. 2. Collection Task Ref_Perf_BckGnd_Task_TR-198_Multi_NE is the same as Collection Task Ref_Perf_BckGnd_Task_TR-198 but it is applied on NE_1 and NE_2. 3. Let N be the number representing 10% of the total number of the lines supported by the NE_1 and NE_2, assumed to be of the same equipment type. 4. Create N tasks of Collection Task Ref_Perf_Task-k_TR-198_Multi_NE on the DCF via the DCF manager. 5. Collection Task Ref_Perf_Task-k_TR-198_Multi_NE is the same as Collection Task Ref_Perf_Task-k_TR-198 but it is applied on NE_1 and NE_2. 6. Activate Collection Task Ref_Perf_BckGnd_Task_TR-198 and the N tasks created in step 2. 7. Wait until a number of collection and upload periods (as configured for the specific parameter groups) pass, then request the DCF manager to show all the data reports, associated to all the collection tasks activated in step 4. 8. Wait until the duration periods of all the collection tasks activated in step 4 expire, then request the DCF manager to show all the data reports, associated to all of those tasks.
Expected Result	1. The DCF MUST be able to collect and upload all the requested parameters for each configured collection period throughout the duration of each of the involved tasks. 2. All the data reports generated by the configured collection/upload scheduling MUST be available on the DCF manager and contain all the parameters configured for collection.

7   DCF load stress tests [Optional]

These tests aim to verify that the DCF fulfills the maximum limits for collection and upload, as declared by the manufacturer. The following tests deal with limits on the number of collected parameters on a single line, the number of concurrent lines under collection, the number of active CollectionTasks.

The reference collection tasks for these tests are defined in 4.4.3.

These tests are beyond the actual verification of TR-198 [5] compliance and are included only as an aid to correctly test the DCF implementation. The compliance to these tests is OPTIONAL for the SUT.

7.1   Max parameter for every line in a collection task test [Optional]

Covered requirements:

Test configuration	Refer to Figure 4 if the DCF is integrated in the NE or Figure 5 if not. Lines belonging to VDSL2 linecards configured as per Section 4.3, point a). Lines belonging to ADSL2/2plus linecards as per Section 4.3, point b). Refer to Section 4.2 on how to avoid the use of huge number of CPEs to configure the needed lines.
Test procedure	1. Create the Max_Params_Single_Line task on the DCF via the DCF manager. 2. Start the Max_Params_Single_Line task. 3. Wait until a number of collection and upload periods (as configured for the specific parameter groups) pass, then request the DCF manager to show all the data reports, associated to the collection task activated in step 2. 4. Wait until the duration period of the collection task activated in step 2 expires, then request the DCF manager to show all the data reports, associated to this task.
Expected result	3. The DCF MUST be able to collect and upload all the requested parameters for each configured collection period throughout the duration of the involved task. 4. All the data reports generated by the configured collection/upload scheduling MUST be available on the DCF manager and contain all the parameters configured for collection.

7.2   Max lines per board collection task test [Optional]

Covered requirements:

Test configuration	Refer to Figure 4 if the DCF is integrated in the NE or Figure 5 if not. Lines belonging to VDSL2 linecards configured as per Section 4.3, point a). Lines belonging to ADSL2/2plus linecards as per Section 4.3, point b). Refer to Section 4.2 on how to avoid the use of huge number of CPEs to configure the needed lines.
Test procedure	1. Create the Max_Lines_Per_Linecard task on the DCF via the DCF manager. 2. Start the Max_Lines_Per_Linecard task. 3. Wait until a number of collection and upload periods (as configured for the specific parameter groups) pass, then request the DCF manager to show all the data reports, associated to the collection task activated in step 2. 4. Wait until the duration period of the collection task activated in step 2 expires, then request the DCF manager to show all the data reports, associated to this task.
Expected result	1. The DCF MUST be able to collect and upload all the requested parameters for each configured collection period throughout the duration of the involved task. 2. All the data reports generated by the configured collection/upload scheduling MUST be available on the DCF manager and contain all the parameters configured for collection.

November 2011   42 of 43

7.3   Max lines for collection task test [Optional]

Covered requirements:

Test configuration	Refer to Figure 4 if the DCF is integrated in the NE or Figure 5 if not. Lines belonging to VDSL2 linecards configured as per Section 4.3, point a). Lines belonging to ADSL2/2plus linecards as per Section 4.3, point b). Refer to Section 4.2 on how to avoid the use of huge number of CPEs to configure the needed lines.
Test procedure	1. Create the Max_Lines_Per_NE task on the DCF via the DCF manager. 2. Start the Max_Lines_Per_NE task. 3. Wait until a number of collection and upload periods (as configured for the specific parameter groups) pass, then request the DCF manager to show all the data reports, associated to the collection task activated in step 2. 4. Wait until the duration period of the collection task activated in step 2 expires, then request the DCF manager to show all the data reports, associated to this task.
Expected result	1. The DCF MUST be able to collect and upload all the requested parameters for each configured collection period throughout the duration of the involved task. 2. All the data reports generated by the configured collection/upload scheduling MUST be available on the DCF manager and contain all the parameters configured for collection.

End of Broadband Forum Technical Report TR-260