TECHNICAL REPORT

TR-403

PON Abstraction Interface for Time-Critical Applications

Issue: 1.0

Issue Date: December 2018

Issue History

Issue Number	**Approval Date**	**Publication Date**	**Issue Editor**	**Changes**
1	3 December 2018	3 December 2018	Kota Asaka, NTT	Original

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

Editor:	Kota Asaka	NTT
Fiber Access Network Work Area Directors:	Marta Seda	Calix
	Wei Lin	Huawei
PON Abstraction Interface for Time-Critical Applications Project Stream Leader:	Kota Asaka	NTT

Table of Contents

List of Figures

List of Tables

Executive Summary

For flexible and agile service adaptation at a low cost in the next-generation optical access system, a new system architecture, which is based on SDN/NFV technologies, should be required from operators' point of view. The most promising way to pursue the new architecture is disaggregating PON functions to functional modules with open interfaces.

This Technical Report defines the time critical interface described in TR-402.

1 Purpose and Scope

1.1 Purpose

The purpose of this Technical Report is to specify the detail specifications for PON abstraction interface as APIs under the project of "PON abstraction interface for time-critical applications". The API specification in this Technical Report contains the format, model, and performance requirements.

1.2 Scope

The scope of this project is to specify the PON abstraction interface for time-critical applications. The project assumes the disaggregation of access systems by using SDN/NFV technologies. However, in order to replace or update time-critical processing functions (e.g., DBA), the software must be replaced or updated with the use of commonly usable (or open) APIs in the OLT.

The following documents define the PON abstraction interface:

- TR-402: Defines the functional modules and interfaces so as to disaggregate a PON function (e.g., DBA) that needs time-critical processing.

- TR-403: Defines the interface specifications for DBA in ITU-T PON. Future revision of this document will include IEEE PON specifications.

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.

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 term “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 behavior is acceptable or even useful, but the full implications need to be understood and the case carefully weighed before implementing any behavior described with this label.
MAY	This word, or the term “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]	*G.989.3 Amd.1*	*40-Gigabit-capable passive optical networks (NG-PON2): Transmission convergence layer specification Amendment 1*	ITU-T	2016
[2]	*G.984.3*	*Gigabit-capable passive optical networks (G-PON): Transmission convergence layer specification*	ITU-T	2014

2.3 Definitions

The following terminology is used throughout this Technical Report.

Time-critical application	The time-critical application is a function in optical access system (e.g., PON), which requires high-speed processing, such as DBA.
PON abstraction interface	The PON abstraction interface is an API that provides flexibility of time-critical application. The time-critical application having algorithm dependent function can be disaggregated to the common behavior part as engine, and the differentiation part as algorithm. The interface between two parts are called the PON abstraction interface, and specified as an API. By using APIs, differentiation/flexibility of time-critical applications can be achieved by updating/replacing algorithm part, which is a software component.

2.4 Abbreviations

This Technical Report uses the following abbreviations:

API	Application Programming Interface
CPU	Central Processing Unit
DBA	Dynamic Bandwidth Assignment
OLT	Optical Line Termination
ONU	Optical Network Unit
PON	Passive Optical Network
TR	Technical Report

3 Technical Report Impact

3.1 Energy Efficiency

This Technical Report has no impact on Energy Efficiency since it is assumed that time-critical applications reside in an OLT.

3.2 IPv6

This Technical Report has no impact on IPv6.

3.3 Security

This Technical Report has no impact on Security.

3.4 Privacy

This Technical Report has no impact on Privacy.

4 Common API specifications

4.1 DBA

This section discusses the API format and structure between the TR-402 engine and CPU.
The API defines the following commands:

Set: Set commands provide set functions from the CPU to the engine.

Get: Get commands retrieve information from the engine.

4.1.1 Format and model

These API formats are considered to apply for ITU-T PON as shown in Table 4-1. This is minimum set of parameters needed for DBA APIs.

Table 4-1 Common API format for DBA

Name	Input arguments (type/content)		Output arguments (type/content)	
	Type	Content	Type	Content
setGrant	uint8	Engine number		
	uint8	PON ID		
	GM_TYPE	Grant message		
getReport			uint8	PON ID
			uint32	DBA cycle number
			uint64	SFC
			uint32	Available BW blocks
			uint32	Num of Alloc IDs
			uint32	Num of ONUs
			PQS_TYPE	PLOAM queue status[32]
			RL_TYPE	Report List[1024]

Data structure	Type	Content
PQS_TYPE	uint16	ONU ID
	uint8	pqs

Data structure	Type	Content
GM_TYPE	uint32	DBA cycle number
	uint32	List size
	GL_TYPE	Grant List[2048]
GL_TYPE	uint16	Alloc ID
	uint16	Allocation size
	uint16	Start time
	uint8	Burst profile
	FL_TYPE	Alloc flags
FL_TYPE	bool	reserved
	bool	reserved
	bool	reserved
	bool	FWI
	bool	End of map
	bool	End of frame
	bool	DBRu flag
	bool	PLOAMu flag

Data structure	Type	Content
RL_TYPE	uint16	Alloc ID
	uint32	Allocated
	uint32	Used
	uint32	Status report

Figure 4-1 shows assumed granting model of DBA. Single API call can allocate multiple allocation structures with multiple BWmaps. The event for DBA to obtain control is not limited to report (BufOcc) reception.

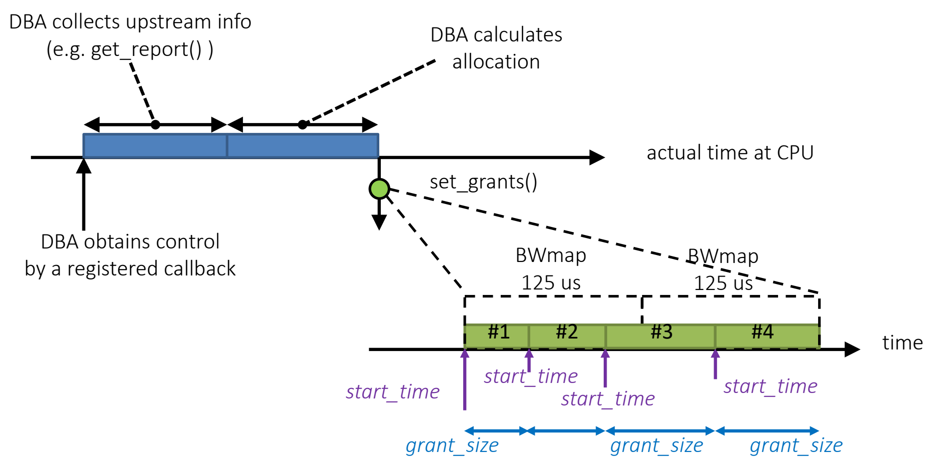

Figure 4-1 Granting model of DBA

4.1.2 API Data structure

4.1.2.1 Arguments for setGrant()

Table 4-2 Data structure for setGrant

Name	Type	Description
Engine number	uint8	From the viewpoint of application (algorithm), this field is used to designate which engine to call this API.
PON ID	uint8	This field is used to designate the target logical PON port.
DBA cycle number	uint32	DBA is basically processed on the cycle basis. This field is used to indicate the timeline of the allocation. This field is expected to be incremented from the previous DBA cycle. Wrap around may occur.
List size	uint32	This field represents the number of elements in GrantList.
Alloc ID	uint16	This field represents the target *Alloc-ID* in ITU-T G.989.3.
Allocation size	uint16	This field represents *GrantSize* in ITU-T G.989.3. The unit of this field is 4-byte block for 2.48832Gbps and 16-byte for 9.95328Gbps.
Start time	uint16	This field is used to indicate *StartTime* in ITU-T G.989.3. The unit of this field is 4-byte block for 2.48832Gbps and 16-byte for 9.95328Gbps.
Burst profile	uint8	This field is used to indicate the burst profile of the upstream. This field corresponds to *BurstProfile* field, described in ITU-T G.989.3, in constructing BWmap.
reserved	bool	This field is reserved.
FWI	bool	This field is used to notify the ONU to wake up during ONU power management process. This field corresponds to Forced wake-up indication (FWI) bit in ITU-T G.989.3.
End of map	bool	This field is used to notify the engine that this element is the last one in the API call.

End of frame	bool	This field is used to notify the engine that this element is the last one in the 125s BWmap.
DBRu flag	bool	This field represents *DBRu* flag in ITU-T G.989.3.
PLOAMu flag	bool	This field represents *PLOAMu* flag in ITU-T G.989.3.

4.1.2.2 Arguments for getReport()

Table 4-3 Data structure for getReport

Name	Type	Description
PON ID	uint8	This field represents the source logical PON port of the status report.
DBA cycle number	uint32	DBA is processed on the cycle basis. This field is used to indicate the timeline of the allocation. This field is expected to be incremented from the previous DBA cycle.
SFC	uint64	This field is used to indicate the timeline of the allocation. This field represents the first frame of the DBA cycle. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
Available BW blocks	uint32	This field represents how many BW blocks are available for whole active *Alloc-ID*s. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
Num of Alloc IDs	uint16	This field represents the number of *Alloc-ID*s included in this callback. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
Num of ONUs	uint16	This field represents the number of active ONUs on which we will receive the PLOAM queue status in this callback. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
Alloc ID	uint16	This field represents *Alloc-ID* of the status report. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
Allocated	uint32	This field represents how many BW bocks are allocated to the *Alloc-ID*. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
Used	uint32	This field represents how many BW blocks are actually used by the *Alloc-ID*. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
Status report	uint32	This field represents *BufOcc* in ITU-T G.989.3 of the *Alloc-ID*. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
ONU ID	uint16	This field represents the ONU ID of the PLOAM queue status. Required to be in Network Byte order defined in Appendix B of IETF RFC791.
pqs	uint8	PLOAM queue status of the corresponding ONU.

4.1.3 Performance requirement

From a viewpoint of service providers, the criteria for determining whether their use case (e.g., a new service) is feasible or not with specified interfaces are essential. Specifically, for time-critical applications, the criteria are performance requirement.

The performance requirement should not depend on a specific sequence or implementation. Therefore, this performance requirement should not include the processing time of an algorithm itself. Additionally, the performance requirement should be defined for each API individually because each API can be called at any time (or may not be called) according to the algorithm.

4.1.3.1 Time definition

In defining the criteria of performance requirement, the scope also needs be defined. In this subsection, the target time scope for the requirement is defined.
As described in Figure 4-2, the call interval is defined as the time between when the algorithm called an API and when the API becomes available for the next call from the perspective of API caller. On the other hand, the processing time in Figure 4-2 is defined as the time between when the algorithm called an API and when the result is reflected. As illustrated in Figure 4-3, call interval may be shorter than processing time.

For callback API as described in Figure 4-4, the call interval is defined as the time between when the algorithm is called back with the API and when the next time the algorithm is called back with the same callback API. On the other hand, the processing time in Figure 4-4 is defined as the time between when the engine started the process of the callback API and when the algorithm is called back. The processing time does not include the duration which an engine waits for an event to trigger the process of the callback API.

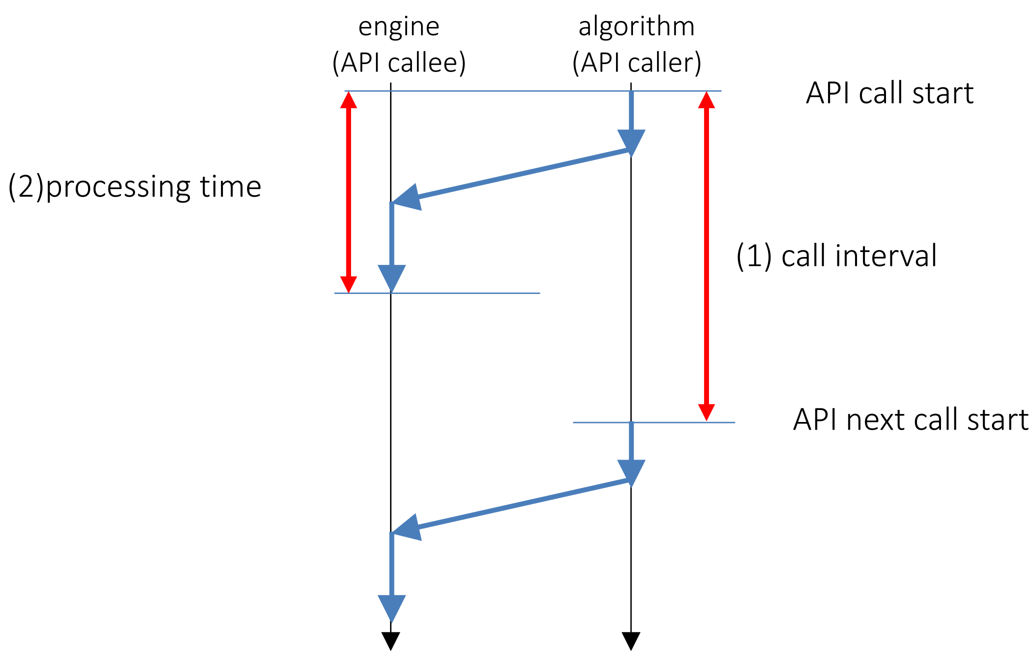

Figure 4-2 Call interval and processing time

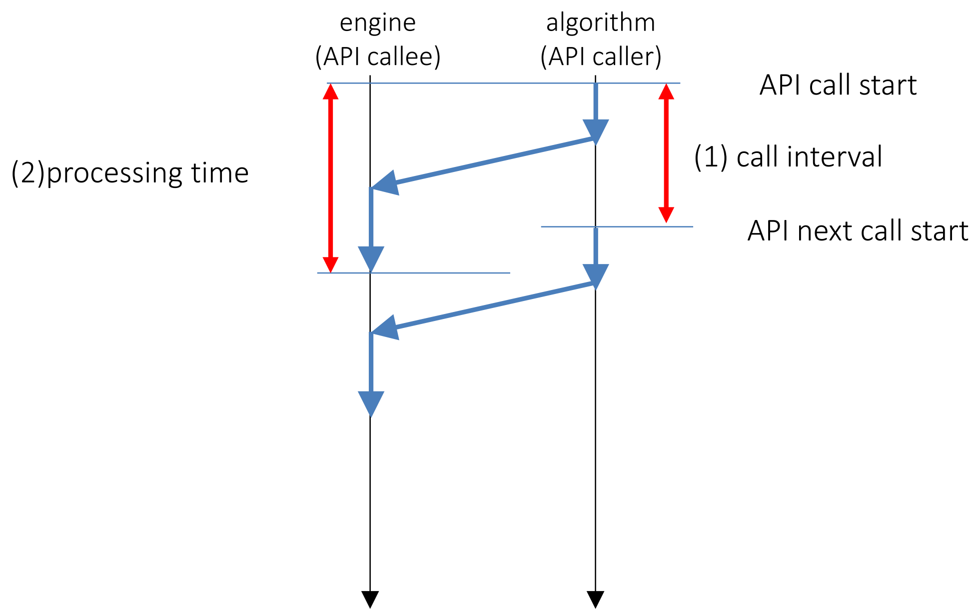

Figure 4-3 Shorter call interval than processing time

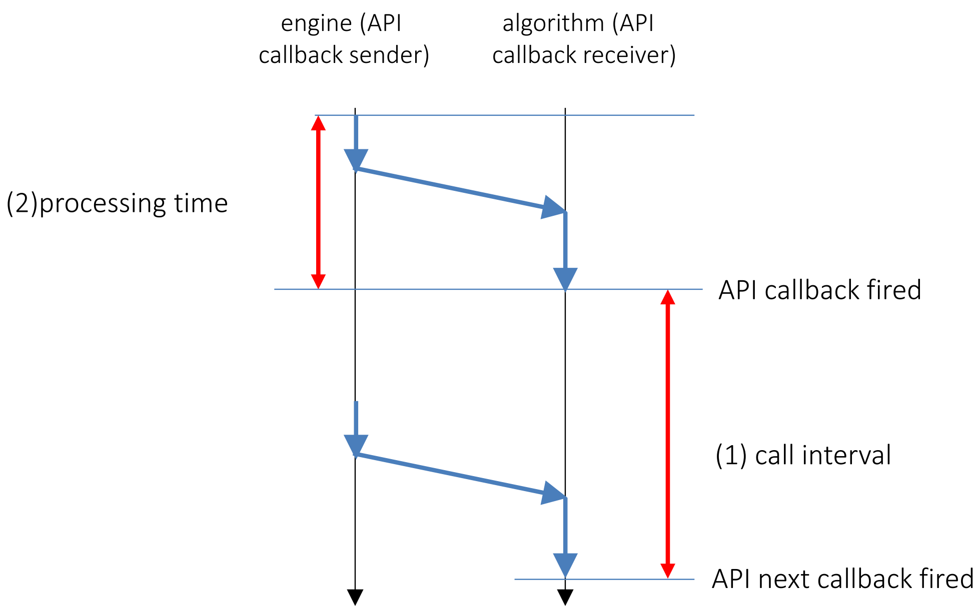

Figure 4-4 Call interval and processing time for callback API

4.1.3.2 Time requirement

As described in TR-402, there are several use cases with different time requirements according to the applications. An API that satisfies a more stringent time requirement provides a broader range of applications. One of the most demanding time requirements for DBA APIs would be the one for mobile fronthaul use case. Considering the convenience in providing various applications for both sides of algorithm and engine, time requirement with multiple levels are useful.

In the mobile world, there are multiple levels on time-related specifications. Specifically, 3GPP TS 38.211 describes several patterns of the slots where a slot is the minimum time unit of scheduling in

5G mobile system. Although the time duration of a subframe is defined as long as 1ms in 5G mobile systems, the number of slots within a subframe can be 16 at the maximum. This is extended in 5G from LTE mobile systems which are based on 1ms-long TTI (transmission time interval) as the minimum time unit of the scheduling.

Table 4-4 shows time requirement classes for DBA APIs which is based on the slot duration in TS 38.211.

For example with Table 4-4, if a developer has to assure some DBA algorithm to be completed within 250 us where the algorithm itself takes 100 us processing time in addition to the API call, class 4 or higher class is required rather than class 3.

The API whose processing time is equal to or less than 1000 us can be classified to class 1. In other words, the API of class 1 MUST NOT exceed the maximum processing time of 1000 us. On the other hand, the API of class 1 MUST be able to be called at least with minimum call interval of 1000 us. The API that can be called with 600-us call interval should be classified to class 1 because it cannot satisfy class 2 requirement. From the viewpoint of the algorithm developer, the API can be called with any longer interval than the minimum call interval. The API call with a shorter interval than the minimum call interval is not guaranteed.

Table 4-4 Time requirement classes

Class	Minimum call interval	Maximum processing time
Class 1	1000 [us]	1000 [us]
Class 2	500 [us]	500 [us]
Class 3	250 [us]	250 [us]
Class 4	125 [us]	125 [us]
Class 5	62.5 [us]	62.5 [us]

4.1.4 Requirement of APIs for time-critical DBA

[R-1] The DBA engine MUST support “setGrant” command as described in Table 4-1 and Table 4-2.

[R-2] The DBA engine that supports “setGrant” MUST satisfy at least one of the classes inTable 4-4.

[R-3] The DBA engine MUST support “getReport” as described in Table 4-1 and Table 4-3.

[R-4] The DBA engine that supports “getReport” MUST satisfy at least one of the classes in Table 4-4.

[R-5] The data structure MUST support the Network Byte order defined in Appendix B of IETF RFC791.

5 APPENDIX A Common API for ITU-T PON and IEEE PON systems (Informative)

IEEE PON system is left for future study. Future versions of this document will include a common API for both ITU-T PON and IEEE PON systems.

End of Broadband Forum Technical Report TR-403