Flexible Access System Architecture (FASA)
White Paper

Ver. 1.0

NTT Access Network Service Systems Laboratories,

NTT Corporation

June 29, 2016
Table of Contents

SUMMARY
1. INTRODUCTION
1.1. Purposes of This Document
1.2. Overview of FASA
1.3. Scope of FASA Application API specification in This Document
1.4. Terms, Definitions, and Acronyms
1.5. Reference
2. Reference Architecture
2.1. Logic Model
2.2. Example of System Configuration
3. Use Cases
3.1. Multi-Service Accommodation by DBA Replacement
3.2. Satisfaction of Requirements Specific to Telecommunications Carrier by Replacement of Power Saving Function
3.3. Satisfaction of Requirements Specific to Telecommunications Carrier by Replacement of Protection Functions
4. Main Functions of Access Network System and Target of FASA Application
5. API specification
5.1. PON Data Signal Processing Function
5.2. PON Access Control Function
5.2.1 DBA
5.3. L2 Data Signal Processing Function
5.4. Maintenance and Operation Function
5.5. PON Multicast Function
5.6. Power-Saving Control Function
5.7. Frequency/Time-of-Day Synchronization Function
5.8. Protection Function
SUMMARY

Because of the diversification of telecommunications usage, in addition to B2C model telecommunications services that are directly provided to end users by telecommunications carriers, B2B2C model telecommunications services are being provided more often to end users by way of various service providers. Also, NTT announced, in 2014, the "HIKARI Collaboration Model,"1 and new services are to be provided through co-creation with various business players. Given these changes in the situation, access network systems also need to be able to cope with various requirements (e.g. bandwidth, cost, and reliability) from service providers, which correspond to the "Middle B"s of B2B2C, and end users flexibly and quickly.

Existing access network systems have been developed as network elements specific to each service, thus access network services are unable to flexibly meet specific requirements. Therefore, it is difficult for conventional access network elements to meet the diversifying requirements of access network services quickly. This will become more important due to changes in business models as well as the increasing variety of telecommunications services. For example, the entire elements must be redesigned even if just one network function needs to be added or revised.

In order to satisfy the diversifying requirements and in order to provide services flexibly and quickly, NTT is advocating the NetroSphere Concept.2 Based on this concept, we are accelerating the research and development of technology that can modularize the functions needed to configure a network and permit their flexible combination as needed. In access networks, NTT has just announced FASA (Flexible Access System Architecture)3 as a new access system architecture.

FASA is based on the NetroSphere concept and modularizes the functions offered by an access network element; FASA allows access network systems to be configured through the flexible combination of modularized functions. The functions, which differ from service to service and/or among telecommunications carriers, are implemented as replaceable software modules (FASA Applications). Such software modules run on the platform (FASA Platform) that provides common interfaces. This configuration enables the functions of access network elements to be added or replaced flexibly in order to satisfy the diversifying requirements quickly and flexibly.

Fig. 1 shows examples of the configurations with modularization based on FASA.

In configuration 1, FASA Applications, which are the functions to be replaced to satisfy service-specific requirements or carrier-specific requirements, are modularized as software.

In configuration 2, the functions inside the FASA Platform (other than the FASA Applications) are also included in the target of software modularization, which realizes access network elements using

2 http://www.ntt.co.jp/news2015/1502/150219a.html
more general-purpose hardware.

This document describes the architecture of FASA and the common interfaces (FASA Application APIs) used in implementing FASA Applications.

Fig. 1. Examples of the configurations with modularization based on FASA
1. INTRODUCTION

1.1. Purposes of This Document

This document describes the FASA (Flexible Access System Architecture) as a new access system architecture that is capable of quickly supporting various requirements from telecommunications carriers and various services and applications—and the FASA Application APIs (Application Programming Interfaces) common interfaces generalized for implementing FASA Applications.

1.2. Overview of FASA

Fig. 1.2.-1 shows existing and future situations of the access network system.

In the current access network system and network element configuration, each function is implemented as a system and network elements dedicated to each service; this results in many restrictions that require redevelopment of the system and/or the whole network element for the addition and/or replacement of functions. In addition, as for the maintenance and operation, spare equipment and maintenance skills specific to each network element are required in this situation. This is why, as shown in this view, it is necessary to enhance the flexibility and the expandability of the access network element and so permit the quick satisfaction of the requirements and services of various telecommunications carriers.

<table>
<thead>
<tr>
<th>Conventional access network system</th>
<th>Requirements</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lots of restrictions in adding-on and/or replacing functions</td>
<td>Highly flexible network system and network element configuration</td>
<td>To accommodate various service levels To add-on/replace functions promptly</td>
</tr>
<tr>
<td>Difficult to drastically reduce cost of dedicated equipment</td>
<td>Highly general-purpose network system and network element configuration</td>
<td>To reduce CAPEX by constructing network systems on demand</td>
</tr>
<tr>
<td>Necessity of maintenance skills and spare equipment for each dedicated system</td>
<td></td>
<td>To reduce OPEX by minimization of spare equipment and simplification of operation procedures</td>
</tr>
</tbody>
</table>

Fig. 1.2.-1. Existing and future situations of the access network system

FASA is new access system architecture and its concept to realize the followings:

1. Access network elements are modularized, instead of developing the access network elements dedicated to specific functions or services.
2. The functions that differ from service to service and/or among telecommunications carriers are realized by software modules with the common interfaces.
3. The dependency among the software modules is minimized and the replaceable software modules run on a platform; therefore, while service quality is maintained, necessary functions are realized flexibly and economically to satisfy the service requirements.
Fig. 1.2.-2 shows a schematic view of modularization based on FASA. As shown in this figure, an access network element based on FASA consists of FASA Applications and the FASA Platform.

"FASA Applications" realize the functions that differ from service to service and/or among telecommunications carriers as software modules with the common interfaces (FASA Application APIs). As FASA Applications are added and/or replaced depending on services, the services with various requirements can be provided quickly and easily.

The "FASA Platform" is a basic component of the access network element that provides the FASA Applications with the FASA Application APIs and, in addition, provides the functions that do not need to be changed with the service or requirement. The FASA Platform realizes each function that forms the FASA Platform by using hardware or software depending on the processing performance and other requirements. Fig. 1.2.-3 shows examples of the configurations with modularization based on FASA.
In configuration 1, the FASA Applications over the FASA Application APIs are the targets of software modularization, while the modularization of each function forming the FASA Platform is out of the scope. For example, in the case of an NG-PON2 system based on this configuration, the NG-PON2 protocol, which is standardized by ITU-T, is implemented on the FASA Platform, and the replacement of NG-PON2 to EPON and other PON protocols is not considered.

In configuration 2, each function that composes the FASA Platform is modularized in addition. Namely, FASA platform is also included in the scope of software modularization.

The FASA Application APIs are commonly used in both configuration 1 and configuration 2.

Fig. 1.2.-4 shows an example wherein the FASA Platform is divided into general-purpose hardware and add-on modules. The access network element shown in the figure is composed of three module categories: "(1) FASA Application," "(2) general-purpose hardware," and "(3) add-on module." Their combination allows the necessary functions to be provided quickly and easily. "(2) general-purpose hardware" has general-purpose telecommunications functions for not only access network elements but also other network elements, which are implemented as general-purpose modules. By using general-purpose hardware, it is possible to avoid the frequent development of network elements from scratch with changes in service requirements. In addition, by using general-purpose modules, it is expected that network element cost will be cut and that the maintenance operations will be simplified through the commonality of spare equipment and the like. "(2) general-purpose hardware" is, for example, some hardware that is not a network element dedicated to the access network system such as general-purpose servers and white box switches. It is equipped with firmware, OS and other software necessary for the hardware, and the middleware for the FASA Application APIs to allow the use of "(1) FASA Application." "(3) Add-on module" implements an optical transceiver and other functions where these are hard to implement as "(1) FASA Application" or to implement on "(2) general-purpose hardware". It is separated from "(2) general-purpose hardware." "(3) add-on module" is some hardware other than "(2) general-purpose hardware" such as an optical transceiver. In the case where a simple optical module is used as "(3) add-on module" to implement a media converter or in similar cases, "(1) FASA Application" will be executed by using the OS, the middleware for the FASA Application APIs for "(1) FASA Application" and other software on "(2) general-purpose hardware." On the other hand, in the case where the MAC chip of a PON is implemented on "(3) add-on hardware" or in other similar cases, "(1) FASA Application" will be executed by using the OS, the middleware for the FASA Application APIs for "(1) FASA Application" and other software on "(3) add-on module." In either case, it is possible to realize an access network with the optimal transmission capacity and the optimal transmission technology by replacing add-on modules and corresponding software according to the service requirements while using the same general-purpose hardware.
1.3. Scope of FASA Application API specification in This Document

This document describes use cases of the FASA-based access network system, system architecture for realizing the use cases, the functions formed as FASA Applications, and the FASA Application APIs. Currently, as for the access network system, because the MAC chip of the PON and other dedicated hardware may be necessary from the point of view of processing performance and the like, the modularization of each function provided on the FASA Platform (configuration 2 in Fig. 1.2.-3) is left out of the scope of this document. Specifically this document focuses on the PON OLT in configuration 1 in Fig. 1.2.-3.

1.4. Terms, Definitions, and Acronyms

FASA-API: The common term of the APIs used for FASA.

FASA Application: The replaceable software modules implemented by using the FASA Application APIs.

FASA Application API: The APIs that connect FASA Applications and the middleware for the FASA Application APIs.

Middleware for the FASA Application APIs: In the FASA Platforms, the software that provides FASA Applications with the FASA Application APIs. The middleware for the FASA Application APIs provides the means for inter-function communication among FASA Applications. The middleware also provides the means for inter-function
communication of FASA Applications and other functions in FASA Platforms. The middleware absorbs any differences among the FASA Platforms.

FASA Platform: A basic component of an access network element that provides FASA Applications with the FASA Application APIs. The component also provides functions that do not depend on the service or requirement because of standardization or the like.

Add-on module: The module that implements a function that is difficult to implement on general-purpose hardware by separating it from general-purpose hardware.

Software module: The module that forms a necessary function in a replaceable unit.

General-purpose hardware: The hardware whose usage is not limited to the access network service, e.g. a general-purpose server and white box switch.

ACK: ACKnowledgement
ACPI: Advanced Configuration and Power Interface
API: Application Programming Interface
ASIC: Application Specific Integrated Circuit
B2B: Business-to-Business
B2B2C: Business to Business to Consumer
BBU: Base Band Unit
BWMap: BandWidth Map
CLI: Command Line Interface
CT: Channel Termination
DBA: Dynamic Bandwidth Assignment
DWA: Dynamic Wavelength Assignment
DWBA: Dynamic Wavelength and Bandwidth Assignment
EPON: Ethernet Passive Optical Network
FASA: Flexible Access System Architecture
FBA: Fixed Bandwidth Assignment
FEC: Forward Error Correction
FTTH: Fiber to the Home
GEM: G-PON Encapsulation Method
G-PON: Gigabit-capable Passive Optical Network
GTC: G-PON Transmission Convergence
IEEE: The Institute of Electrical and Electronics Engineers
IF: Interface
IPv6: Internet Protocol Version 6
ITU-T: International Telecommunication Union Telecommunication Standardization Sector
L2: Layer 2
L3: Layer 3
LTE: Long Term Evolution
MAC: Media Access Control
MFH: Mobile Front Haul
MLD: Multicast Listener Discovery
NBI: Northbound Interface
NE-OpS: Network Element-Operation System
NG-PON2: Next Generation Passive Optical Network 2
NSR: Non-Status Reporting
OAM: Operation Administration and Maintenance
ODN: Optical Distribution Network
OLT: Optical Line Terminal
OMCI: ONU Management and Control Interface
ONU: Optical Network Unit
OS: Operating system
OSS: Operators Management System
OSU: Optical Subscriber Unit
PHY: Physical Layer
PLOAM: Physical Layer OAM
PON: Passive Optical Network
PS: Power Save
PTP: Precision Time Protocol
QoS: Quality of Service
RRH: Remote Radio Head
SBI: South Bound Interface
SD: State Diagram
SDK: Software Development Kit
SFC: Super Frame Counter
SNI: Service Node Interface
SNMP: Simple Network Management Protocol
SP conversion: Serial-to-Parallel conversion
SR: Status Reporting
SyncE: Synchronous Ethernet
TBD: To Be Determined
TC: Transmission Convergence
TDD: Time Division Duplex  
TDM: Time Division Multiplexing  
ToD: Time of Day  
TRx: Transceiver  
UE: User Equipment  
UNI: User Network Interface  
VLAN: Virtual LAN (Local Area Network)  
XGEM: XG-PON Encapsulation Method  
XGTC: XG-PON Transmission Convergence  
XG-PON: 10 Gigabit Capable Passive Optical Network  
WDM: Wavelength Division Multiplexing  
WDM/TDM-PON: Wavelength Division Multiplexing and Time Division Multiplexing Passive Optical Network

1.5. Reference

2. Reference Architecture
2.1. Logic Model
Fig. 2.1.-1 shows the logic model; the example is that of the OLT. The controller and external equipment are not included in the OLT, but are shown merely to illustrate the communications with the FASA Application APIs. The logic model is composed of the FASA Applications and the FASA Platform that provides the FASA Applications with the FASA Application APIs. The FASA Platform includes the middleware for FASA Application APIs. The middleware for the FASA Application APIs absorbs the differences among vendors and types of hardware and software forming the FASA Platform. The FASA Application API set that is not
dependent on a vendor or service type is specified on the middleware for the FASA Application APIs, and the FASA Applications are replaced to provide the functions necessary for each service or for each telecommunications carrier. Communications among the FASA Applications and communications between FASA applications and the controller or the like are executed by way of the middleware for the FASA Application APIs.

The FASA Application API set is the group of common APIs used by the FASA Applications, and APIs necessary for each FASA Application are selected as needed from the API set.

The controller is the operation system to manage the OLT such as NE-OpS, and control signals from the controller are used for the communications with the SBI application by way of the FASA Application APIs. The control signals for the communications with the SBI application are terminated at the application for maintenance and operation by way of the FASA Application APIs.

The external equipment is, for example, the BBU of a mobile system or some other OLT; this is external equipment that communicates with the FASA Applications by way of the FASA Application APIs. In this figure, the external equipment (BBU) is communicating with a DBA application.

In this figure, communication with functions in the FASA Platform below the middleware for the FASA Application APIs, communication among the applications, and the communication between the maintenance and operation application and other applications are shown with red, green, and orange arrows, respectively.
2.2. Example of System Configuration

Fig. 2.2.-1 takes NG-PON2 OLT as an example and shows an example of the configuration of an access network element based on FASA. The figure shows an example where the FASA Platform is composed of two or more sets of hardware (NG-PON2 box, white box switch). The NG-PON2 box and the white box switch are connected using a standard protocol such as Ethernet. The addition and/or replacement of functions is performed by adding and/or replacing FASA Applications on the FASA Application APIs of the FASA Platform. The middleware for the FASA Application APIs is the software that absorbs the difference in the hardware and the software arising from the differences among vendors and/or service types.
3. Use Cases

As use cases of FASA, the multi-service accommodation and the satisfaction of the requirements specific to a telecommunications carrier by a FASA Application replacement are shown. The first example considers a PON system where FASA Applications are replaced to support diverse services with extremely different requirements such as mobile, residential, and business. Section 3.1 describes a DBA use case, which is a typical example of the above-mentioned.

The latter example considers the provision of access network elements equipped with a desired function by replacing FASA Applications as necessary to support functions whose usage is specific to different telecommunications carriers; not all combinations are implemented in advance. Sections 3.2 and 3.3 describe the use cases of power-saving and protection, which are typical of the above-mentioned example. Fig. 3.-1 shows a schematic view of the satisfaction of the requirements specific to telecommunications carriers by FASA Application replacement.
3.1. Multi-Service Accommodation by DBA Replacement

It is expected that the current PON system, mainly dedicated to FTTH, will be effectively used for multi-service accommodation for users such as mobile, residential, and business. However, the requirements for each service are extremely different. In particular, the maximum delay tolerance with respect to the data signal of the mobile front haul (MFH) is specified more strictly than, for example, conventional Internet access services. Therefore, DBA, which assigns the upstream bandwidth of the PON, needs to follow strict delay specifications. In addition, as for the application of the PON to MFH, it is easily assumed that the requirements will change as a result of the advances in the technology such as the fifth generation (5G) mobile communications, of advances in the standards, and so forth, so that system development must follow such changes in a timely manner. FASA replaces FASA Applications to provide the DBA suitable for each service in a timely manner.

Section 3.1 provides a comprehensive analysis of DBA use cases, and specifies the APIs involved with the addition and/or replacement of future services.

3.1.1. DBA for MFH

DBA is the process that assigns the upstream bandwidth of each ONU (assigned time slot), and can be classified into SR-DBA, which dynamically assigns the bandwidth to each ONU based on the report (the request for bandwidth assignment) from the ONU, and NSR-DBA, which does not use a report from the ONU. SR-DBA can perform assignment with high bandwidth efficiency according to the report from the ONU. However, the control signal exchange, which makes a round trip between
the OLT and the ONU, takes time. Therefore, it is difficult to satisfy MFH requirements. Consequently, the optical-mobile cooperative DBA, which acquires the information necessary for assigning bandwidth from external equipment, and the NSR-DBA, which assigns bandwidth based on traffic forecasts, have been proposed [5-7].

### 3.1.1.1. Optical-Mobile Cooperative DBA for MFH

Fig. 3.1.1.1.-1 shows the configuration and sequence of the optical-mobile cooperative DBA for MFH. In the optical-mobile cooperative DBA for MFH, the OLT cooperates with the BBU to assign bandwidth while satisfying the delay required for MFH. In particular, instead of using the report from the ONU based on the upstream packet arriving from RRH, the optical-mobile cooperative DBA uses the wireless resource information equivalent to the upstream transmission control information generated by BBU for each UE. The upstream bandwidth assignment (the upstream bandwidth and the timing of each ONU) is calculated by the DBA application or the external equipment according to the wireless resource information [5]. This operation eliminates the waiting time at the ONU, which ranges from the arrival time of the upstream signal from UE to the launch time of the signal from ONU to a PON section, and thus reduces the delay. In the case of this DBA, the API must enable linkage with the external equipment.

![Fig. 3.1.1.1.-1 Optical-mobile cooperative DBA for MFH](image)

### 3.1.1.2. NSR-DBA for MFH

Fig. 3.1.1.2.-1 overviews an example of two types of NSR-DBA for MFH. Here, NSR-DBA (which offers fixed bandwidth assignment) removes the roundtrip delay of the control signal generated by SR-DBA and achieves the bandwidth assignment that satisfies the delay required for MFH. In particular, based on the statistical data of the traffic and the traffic pattern, NSR-DBA is used to reduce the excessively assigned bandwidth generated with FBA, and the efficiency of the whole system is enhanced through the shared use with residential systems.
The first NSR-DBA uses the statistical data of traffic to estimate the TDD cycle of the TDD-type mobile system and executes bandwidth assignment according to the TDD cycle [6]. This DBA needs an API that enables the use of the statistical traffic information with sufficient sampling speed for estimating the TDD cycle.

The second NSR-DBA focuses on the fact that there is similarity in traffic pattern if observed using time spans from one day to one week, and estimates the necessary bandwidth for the next traffic period based on the statistical traffic information [7]. This DBA needs an API that enables the use of statistical traffic data gathered over long periods.

3.1.2. SR-DBA for Residential Use

Fig. 3.1.2.-1 shows the configuration and sequence of SR-DBA for residential use. In this SR-DBA, based on the request information collected from each ONU, dynamic bandwidth assignment is executed for each ONU. Therefore, this DBA needs an API that collects the request information from each ONU.
3.1.3. DBA for Business Use

TBD

3.1.4. API of DBA and Functional Block

Based on what is described above, Fig. 3.1.4.-1 shows a schematic view of the API of DBA and the functional block for realizing each of the use cases. In addition, Fig. 3.1.4.-2 shows the definition of each functional block (the policy determination part, the assignment calculation part, the cooperative control part, the traffic monitor, the report processing part, the grant processing part) in Fig. 3.1.4.-1.

---

**Fig. 3.1.4.-1. A schematic view of the API of DBA and the functional blocks**

Use Cases 1 and 2 in Fig. 3.1.4.-1 show the location of the API with configuration for the case wherein the entire DBA (the policy determination part and the assignment calculation part) is...
implemented as a FASA Application in order to flexibly implement DBA with functions such as linkage to external equipment. The SR-DBA for residential use in Use Case 3 of Fig. 3.1.4.-1 shows the location of the API with configuration for the case of ") Implementing a Partial DBA Function as Software" where a part of only the DBA (the policy determination part) is implemented together with the MAC chip etc. of the conventional OLT in addition to "(a) Implementing the Whole DBA Function into Software" where the whole DBA is implemented as a FASA Application.

In order to handle the above-mentioned use cases, the FASA Platform needs to have a cooperative control function and a traffic monitor function, in addition to report processing function and grant processing function, which are specified in the PON standard. FASA Platform also needs to have APIs for collecting the cooperative information, the traffic amount information, the requests, and the parameters for policy determination, and assigned amount, the transmission start time, and the parameters for bandwidth calculation.

![Fig. 3.1.4.-2. Definitions of the functional parts](image)

3.2. Satisfaction of Requirements Specific to Telecommunications Carrier by Replacement of Power Saving Function

TBD

3.3. Satisfaction of Requirements Specific to Telecommunications Carrier by Replacement of Protection Functions

The requirements of telecommunications carriers include the high access network availability. For example, a protection function is essential to reduce the interruption time of the service in case of the
equipment failure. In addition, to provide service reliability during software updates on access network elements as well as migration to new hardware with enhanced processing functions, it is necessary to execute the remedial operations without interrupting the service; therefore, a protection function is critical.

As the access network elements and the network configurations connected to them differ among telecommunications carriers, it is assumed that some protection should be performed differently or in accordance with different policies; therefore, in FASA, an API for protection is specified.

4. Main Functions of Access Network System and Target of FASA Application

The main functions of the access network system and the target of the FASA applications are shown in Table 4.-1. First of all, the main functions of the access network system are described below.

The PON data signal processing function is a group of functions that process the data signals for incoming/outgoing transmissions with the ONU, and so includes functions for the PON frame composition/decomposition and FEC.

The PON access control function is a group of control functions for controlling the incoming/outgoing transmission of the data signals described above, and so includes dynamic bandwidth assignment, registration and authentication of the ONU.

The L2 data signal processing function is a group of functions that forward and process the data signals between the PON-side port and the SNI-side port, and so includes learning the MAC address, VLAN control, QoS control, and traffic monitoring.

The maintenance and operation function is a group of functions for smoothly maintaining and operating the service with the help of the access network element, and so includes functions to configure the settings of the ONU and OLT (the OSU and switch), software updates, management of the access network elements and services, monitoring function for confirming the normality of functions inside the network element, and testing function for investigating the scope and/or the cause of failure. In addition, the maintenance and operation function provides a mechanism to cooperate with the maintenance and operation system which realizes management of multiple access network elements, thus realizes smooth maintenance and operation even from a remote site.

The PON multicast function is a group of functions that forwards the multicast stream received from the SNI-side port to appropriate users, and so includes the identification and distribution of multicast streams and a function to prepare the filter setting of the ONU.

The power-saving control function is a group of functions for reducing the power consumption of the ONU and OLT, which, in addition to the power-saving functions specified by standards, includes a function that minimizes the influence over the service by cooperating with the traffic monitor while acquiring the maximum efficiency in terms of power-saving.
The frequency/Time-of-Day synchronization function is a group of functions for providing the equipment under the ONU with accurate frequency synchronization and time-of-day synchronization. They include a function to synchronize the ONU’s own real-time clock with the master device and a function that uses the PON frame to notify the time-of-day information to the ONU.

The protection function is a group of functions for continuing a service by switching and/or taking over from the working system to the backup system upon the detection of a failure (assumes redundant configuration). The redundancy is provided by deploying two or more devices. This function group includes a function that detects switching trigger and/or executes the switching process. In addition, when a failure is detected or when manual switching is performed, the protection function does not stop the complete service but lets the service continue in restricted operation.
### Table 4-1 The Main Functions of Access Network System and the Target of FASA Applications

<table>
<thead>
<tr>
<th>Functional group</th>
<th>Function</th>
<th>FASA application: Examples of the extension by using FASA application API FASA platform. Functionality provided by FASA platform</th>
<th>FASA application: FASA platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>FN1 (face-to-face processing functions)</td>
<td>Basic function</td>
<td>Decides and specifies the frame header conforming to the given standard. Source: RFC 2119, for example.</td>
<td>Platform</td>
</tr>
<tr>
<td>FN2 (access control functions)</td>
<td>ONU operation functional authentication</td>
<td>Processes the ONU frame conforming to the given standard (e.g., IEEE 802.3, IEEE 802.11, etc.)</td>
<td>Platform</td>
</tr>
<tr>
<td></td>
<td>CER</td>
<td>CER conforms to the IEEE 802.3 standard (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>CoP</td>
<td>CoP supports the standard (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>Traffic management</td>
<td>Supports the traffic management (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>VLAN management</td>
<td>Supports the VLAN management (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>Monitor</td>
<td>Supports the monitor function (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>Maintenance and operation function</td>
<td>Maintenance and operation function</td>
<td>The maintenance and operation function conforming to the given standard (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
</tr>
<tr>
<td></td>
<td>Maintenance and operation function</td>
<td>Maintenance and operation function</td>
<td>The maintenance and operation function conforming to the given standard (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
</tr>
<tr>
<td></td>
<td>Maintenance and operation function</td>
<td>Maintenance and operation function</td>
<td>The maintenance and operation function conforming to the given standard (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>Test function</td>
<td>The test function (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
</tr>
<tr>
<td></td>
<td>Power-saving function</td>
<td>Power-saving function</td>
<td>The power-saving function (e.g., IEEE 802.3, IEEE 802.11, etc.).</td>
</tr>
<tr>
<td></td>
<td>Frequency-domain synchronization function</td>
<td>Frequency-domain synchronization function</td>
<td>The synchronization method conforming to the given standard (e.g., IEEE 1394, etc.).</td>
</tr>
<tr>
<td></td>
<td>Protection function</td>
<td>Protection function</td>
<td>The protection function (e.g., IEEE 1394, etc.).</td>
</tr>
</tbody>
</table>
The next part describes the concepts and examples that examine whether each function should be implemented as a FASA Application or should be implemented in the FASA Platform.

Among the functions shown in Table 4.-1, those that depend on the service and those that need modifications to satisfy a specific requirement of a telecommunications carrier should be implemented as FASA Applications. On the other hand, the functions that are unlikely to be modified for some reasons, e.g. because they are standard-specific, should be implemented on the FASA Platform.

In Table 4.-1, for example, it is shown that the PON data signal processing function is implemented in the FASA Platform. In order to conform an access network element to the ITU-T G.989 series that support 40 Gbit/s rate, it is necessary to follow the standard in implementing the basic PON data signal processing functions such as PON frame composition/decomposition, frame encryption, and forward error correction (FEC). In addition, as these basic functions are common regardless of the service, they should be implemented in the FASA Platform.

As another example, Table 4.-1 shows the case where the DBA function included in the PON access control function is implemented as a FASA Application in order to satisfy service-specific requirements. As described in Section 3, there are some cases where low delay is essential and other cases where bandwidth must be efficiently assigned to the users depending on the service to provide. In order to satisfy the requirements that differ from service to service, it is preferable to separate, as FASA Applications, the procedure and the policy for bandwidth assignment from standard processes (conversion to BWmap format etc. as specified in the standard).

Furthermore, even if the service is for residential use, it is reasonable, for example, that the policy of coping with heavy users may depend on the telecommunications carrier, which means different fairness policies. Specifically, one telecommunications carrier might impose some particular fairness restraint on one PON branch; while another telecommunications carrier performs rough fairness control in at the aggregation switch level. Thus, they have their own QoS requirements to be satisfied.

As described above, FASA aims to fulfil different requirements by replacing FASA Applications; therefore, a means for replacing FASA Applications is required. On the other hand, how to realize the replacement depends on the telecommunications carrier and/or the operation. For example, when the file transfer protocol adopted by an existing maintenance and operation system of a telecommunications carrier is TFTP (Trivial File Transfer Protocol), TFTP is to be equipped; on the other hand, when SFTP (SSH File Transfer Protocol) is used from outside of the maintenance and operation system for service upgrade, SFTP is to be equipped. Moreover, it is assumed that a discussion on standards with respect to these interfaces among access network elements and controllers should be advanced, which means that some consideration should be taken also as the
additions to and changes of the interfaces to follow the development of such standards. This is why
the functions that require customization in accordance with other systems connected to the access
network elements and with its operation are also to be implemented, in Table 4.-1, as FASA
Applications.

Further, FASA is assumed to provide protection by not only fully duplicating the FASA Platform
but also duplicating only a part of the FASA Platform. Examples include, the case where the FASA
Platform is equipped with an optical switch to provide PON protection, the case where two or more
wavelengths are used in one PON branch to realize wavelength protection, the case where only
the switch is duplicated, and the case where the above-mentioned cases are combined; more cases can be
conceived. By implementing the protection function as a FASA Application, it becomes possible to
choose preferable redundant configurations and, furthermore, it becomes possible easily to prepare
various redundant configurations by reusing applicable portions.

5. API specification
The API is, regardless of what is implemented as a FASA Application, a common API set. In other
words, the developer selects which ones to use from the common API set when implementing a
FASA Application. Although the scenarios described in this section are mainly concerned with PON
systems based on ITU-T recommendations, the scope of API includes access network systems
compliant with other PON standards such as those of IEEE.

5.1. PON Data Signal Processing Function
TBD

5.2. PON Access Control Function
5.2.1. DBA
Table 5.2.1.-1 shows the API list for each use case of DBA (the optical-mobile cooperative DBA
for MFH, the NSR-DBA for MFH, the SR-DBA for residential use (the DBA function fully
implemented as a FASA application), and the SR-DBA for residential use (the DBA function
partially implemented as a FASA application)). The column "Use cases" in the table shows the
correspondence between use cases and their APIs. The APIs with a check in each column from "1," "2," "3a," and "3b" in "Use cases" are used, respectively, for the optical-mobile cooperative DBA for
MFH, the NSR-DBA for MFH, the SR-DBA for residential use (the DBA function fully implemented as a
FASA application), and the SR-DBA for residential use (the DBA function partially implemented as
a FASA application). This document assumes a DBA application with event-driven architecture.

The API is, regardless of the FASA Application (the DBA application), a common API set. In
other words, the developer selects which ones to use from the common API set in order to implement
DBA application. For example, "get_AllocationDbaConfig," "set_GrantSize," and "set_GrantSizeAndStartTime" are commonly used by DBA applications, while "get_CooperationConfig," "get_DataSize," "get_Traffic," "get_AllocUnitTraffic," and "get_CalcConfig" are used by a specific DBA application only. The API "get_AllocationDbaConfig," which is commonly used, acquires, from the reply data, the information necessary for each DBA application. The API "set_GrantSizeAndStartTime" is used to control the transmission start time of ONU via the FASA Application. On the other hand, when the API "set_GrantSize" without the start time parameter is used, the FASA platform determines the transmission start time.

In the table, "allocation unit" represents the logical path selected to assign the bandwidth with the DBA. Allocation index represents the indices of the allocation target. Config index are the indices of the setting parameters. Monitoring index are the indices of the measurement target of the traffic monitor. Parameters for policy determination are the parameters used in the calculations of the policy determination part. Parameters for assignment calculation are the parameters used in the calculations in the bandwidth calculation part. In the table, the items tagged <Details to be determined>, the correspondence between an index and a setting parameter, the contents of a parameter for policy determination, and the contents of a parameter for assignment calculation are items to be examined in the future.
<table>
<thead>
<tr>
<th>API</th>
<th>Input</th>
<th>Output</th>
<th>Overview</th>
<th>Use cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>set_init</td>
<td>Xname</td>
<td>Xname</td>
<td>The callback function called at initialization.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>set_T6Proc</td>
<td>Xname</td>
<td>Xname</td>
<td>The callback function called periodically if a specific condition is satisfied (e.g., resource usage).</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>set_ConfigUpdated</td>
<td>Xname</td>
<td>Xname</td>
<td>The callback function called when the configuration is updated.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>set_Finalized</td>
<td>Xname</td>
<td>Xname</td>
<td>The callback function called when the finalization is completed.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_DBConfig</td>
<td>Xname</td>
<td>Xname</td>
<td>The function to collect DBA config information for specific purposes (e.g., monitoring, etc.).</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_AlocationDBConfig</td>
<td>Allocation_index, Xname</td>
<td>Allocation_index, Xname</td>
<td>The function to collect the DBA config information of the allocation unit.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_CooperationConfig</td>
<td>Cooperation_index</td>
<td>Xname</td>
<td>The function to collect the cooperation information in the cooperation control part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_TrafficMonitorConfig</td>
<td></td>
<td>TrafficMonitorConfig</td>
<td>The function to collect the configuration of a traffic monitor (e.g., operation mode).</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_C埇Config</td>
<td>Cooperation_index</td>
<td>Xname</td>
<td>The function to collect the cooperation information in the cooperation control part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_ReportConfig</td>
<td></td>
<td>Xname</td>
<td>The function to collect the report configuration in the report process part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_OemConfig</td>
<td></td>
<td>Xname</td>
<td>The function to collect the configuration in the OEM process part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_AlocationDBConfig</td>
<td>Allocation_index, Xname</td>
<td>Allocation_index, Xname</td>
<td>The function to collect the configuration of the allocation unit.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_CooperationConfig</td>
<td>Cooperation_index</td>
<td>Xname</td>
<td>The function to collect the cooperation information in the cooperation control part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_TrafficMonitorConfig</td>
<td></td>
<td>TrafficMonitorConfig</td>
<td>The function to collect the configuration of a traffic monitor (e.g., operation mode).</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_ReportConfig</td>
<td></td>
<td>Xname</td>
<td>The function to collect the report configuration in the report process part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_OemConfig</td>
<td></td>
<td>Xname</td>
<td>The function to collect the configuration in the OEM process part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_AlocationDBConfig</td>
<td>Allocation_index, Xname</td>
<td>Allocation_index, Xname</td>
<td>The function to collect the configuration of the allocation unit.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_CooperationConfig</td>
<td>Cooperation_index</td>
<td>Xname</td>
<td>The function to collect the cooperation information in the cooperation control part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_TrafficMonitorConfig</td>
<td></td>
<td>TrafficMonitorConfig</td>
<td>The function to collect the configuration of a traffic monitor (e.g., operation mode).</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_ReportConfig</td>
<td></td>
<td>Xname</td>
<td>The function to collect the report configuration in the report process part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_OemConfig</td>
<td></td>
<td>Xname</td>
<td>The function to collect the configuration in the OEM process part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_AlocationDBConfig</td>
<td>Allocation_index, Xname</td>
<td>Allocation_index, Xname</td>
<td>The function to collect the configuration of the allocation unit.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_CooperationConfig</td>
<td>Cooperation_index</td>
<td>Xname</td>
<td>The function to collect the cooperation information in the cooperation control part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_TrafficMonitorConfig</td>
<td></td>
<td>TrafficMonitorConfig</td>
<td>The function to collect the configuration of a traffic monitor (e.g., operation mode).</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_ReportConfig</td>
<td></td>
<td>Xname</td>
<td>The function to collect the report configuration in the report process part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>get_OemConfig</td>
<td></td>
<td>Xname</td>
<td>The function to collect the configuration in the OEM process part.</td>
<td>✔️ ✔️ ✔️</td>
</tr>
</tbody>
</table>
5.2.1.1. Optical-mobile Cooperative DBA for MFH

In Table 5.2.1.-1, the API checked for Use Case 1 is an example of the API of the optical-mobile cooperative DBA for MFH. For example, the API "get_CooperationConfig" acquires the operation cycle and the operation mode of the cooperative control part, the name of the cooperative control system, version, and so forth. The API "get_DataSize" is called every time bandwidth assignment is triggered. This API can acquire either the data size of a specified allocation index individually or those of a specific range of allocation indices at a time.

5.2.1.2. NSR-DBA for MFH

In Table 5.2.1.-1, the API checked for Use Case 2 is an example of the API used by the NSR-DBA for MFH. For example, "get_Traffic" and "get_AllocUnitTraffic" are called in every bandwidth assignment cycle. The API "get_Traffic" measures the traffic flows of multiple OLTs’, OSUs’, or CTs’ while "get_AllocUnitTraffic" measures the traffic flows in terms of allocation unit. For example, "get_AllocUnitTraffic" measures all traffic flows or specific traffic flows identified by the designated allocation indices.

5.2.1.3. SR-DBA for Residential (DBA Function Fully Embodied as a FASA Application)

In Table 5.2.1-1, the API checked for Use Case 3a is an example of the API used by the SR-DBA for residential use (the DBA function fully implemented as a FASA application). For example, "get_ReportSize" is called every time bandwidth assignment is triggered. The API "get_ReportSize" acquires either the reported value of a specified allocation index individually or those of the specified range of allocation indices at a time.

5.2.1.4. SR-DBA for Residential (DBA Function Partially Embodied as a FASA Application)

In Table 5.2.1.-1, the API checked for Use Case 3b is an example of the API used by the SR-DBA for residential use (the DBA function partially implemented as a FASA application). In the API "get_CalcConfig," the parameter "Config information" represents, for example, the maximum bandwidth, or the assured bandwidth of the allocation unit. In the API "get_ParameterForPolicyMaking", the parameter for policy determination is, for example, the history of the requests or assigned amounts.

5.3. L2 Data Signal Processing Function

TBD
5.4. Maintenance and Operation Function
   TBD

5.5. PON Multicast Function
   TBD

5.6. Power-Saving Control Function
   TBD

5.7. Frequency/Time-of-Day Synchronization Function
   TBD

5.8. Protection Function
   TBD
Note: In the event of any discrepancy between this translated document and the Japanese original, the original shall prevail.