新アクセスシステムアーキテクチャ(FASA)
ホワイトペーパー

Ver. 2.0

NTT アクセスサービスシステム研究所

2017/02/28

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

改版履歴

版数	日付	内容
1.0	2016-5-31	作成
2.0	2017-2-28	関連技術，API，各機能について更新

本資料はNTTアクセスサービスシステム研究所が提案するFASAコンセプトについて記したものであり，将来における設備導入，サービス展開を何ら確約するものではない．

また，本資料に記載された内容は，予告なく変更されることがある．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

目次

要旨
1. はじめに
1.1. 本資料の目的
1.2. FASA 概要
1.3. 本資料における FASA アプリケーション API の規定範囲
1.4. 用語と定義及び略語
1.5. 参考文献
2. 参照アーキテクチャ
2.1. 論理モデル
2.2. システム構成例
3. ユースケース
3.1. DBA の入替によるマルチサービス収容の実現
3.2. 省電力機能の入替による通信事業者独自仕様の実現
3.3. プロテクション機能の入替による通信事業者独自仕様の実現
4. アクセスシステムの主要機能と FASA アプリケーション化の対象
5. API 規定
5.1. PON 主信号処理機能
5.2. PON アクセス制御機能
5.2.1 DBA
5.3. L2 主信号処理機能
5.4. 保守運用機能
5.5. PON マルチキャスト機能
5.6. 省電力制御機能
5.7. 周波数/時刻同期機能
5.8. プロテクション機能

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

要旨

通信の用途や利用形態の多様化に伴い、通信事業者が通信サービスをユーザに直接提供する B2C 型通信サービスに加えて、様々なサービス事業者を介して通信サービスを提供する B2B2C 型通信サービスの比重が、大きくなってきている。NTT においても、2014 年に「光コラボレーションモデル」[1]を発表し、様々なビジネスプレイヤとの共創により新たなサービスを提供していくこととしている。このような状況の変化に伴い、B2B2C の「ミドル B」に相当するサービス事業者やエンドユーザからの多様な要求(帯域、遅延、コスト、信頼性やそれらの粒度等)に対して、アクセスシステムにおいても柔軟かつ迅速に対応可能とする必要がある。

従来はこのような個々の要望に対して柔軟に対応するというサービス形態ではなかったこともあり、アクセスシステムはサービス毎の専用装置として開発が行われていた。このため従来の装置では、今後ますます重要となると思われる、サービス毎に種類の異なる要求への対応や、通信サービスのビジネスモデルの変化により、多様化する要求に対応するための機能の追加や入替には、装置全体の再開発が必要であるなど、迅速な対応が困難な状況であった。

このような多様な要求に対応し、柔軟かつ迅速なサービス提供を実現するために、NTT では NetroSphere 構想[2]を提唱しており、ネットワークを構成する機能を部品化し、必要に応じて自由自在に組み合わせて構成する技術の研究開発に取り組んでいる。アクセスネットワークに関しても、同様のコンセプトに基づく新アクセスシステムアーキテクチャ FASA (Flexible Access System Architecture)[3]を発表したところである。

FASA では NetroSphere 構想の考え方に基づき、アクセスネットワーク装置の有する各機能を部品化し、それらの柔軟な組み合わせを可能とする構成を提案している。サービス毎あるいは通信事業者毎に異なる機能は、入替可能なソフトウェア部品(FASA アプリケーション)で実現し、汎用化した入出力インタフェースを提供する基盤(FASA 基盤)上でそれらのソフトウェア部品を動作させる。このような構成により、柔軟な機能の追加や入替を実施可能とし、多様な要求に迅速かつ柔軟に対応するアクセスネットワークを実現する。

図 1 に、FASA による部品化の構成例を示す。

構成 1 は、サービス毎あるいは通信事業者毎に異なる要求に応えるために入替が必要な機能(FASA アプリケーション)をソフトウェア部品化した構成である。

構成 2 は、FASA アプリケーション以外(FASA 基盤)の機能もソフトウェア部品化の対象とすることで、より汎用的なハードウェアを用いてアクセスネットワーク装置を実現する構成である。

本資料では、FASA のアーキテクチャについて説明するとともに、FASA アプリケーション

[1] http://www.ntt.co.jp/irdoc/kaijishiryo/pdf/2014/008.pdf
[2] http://www.ntt.co.jp/news2015/1502/150219a.html
[3] http://www.ntt.co.jp/news2016/1602/160208a.html

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

を実装するための汎用化した入出力インタフェース(FASA アプリケーション API)についても説明をする。

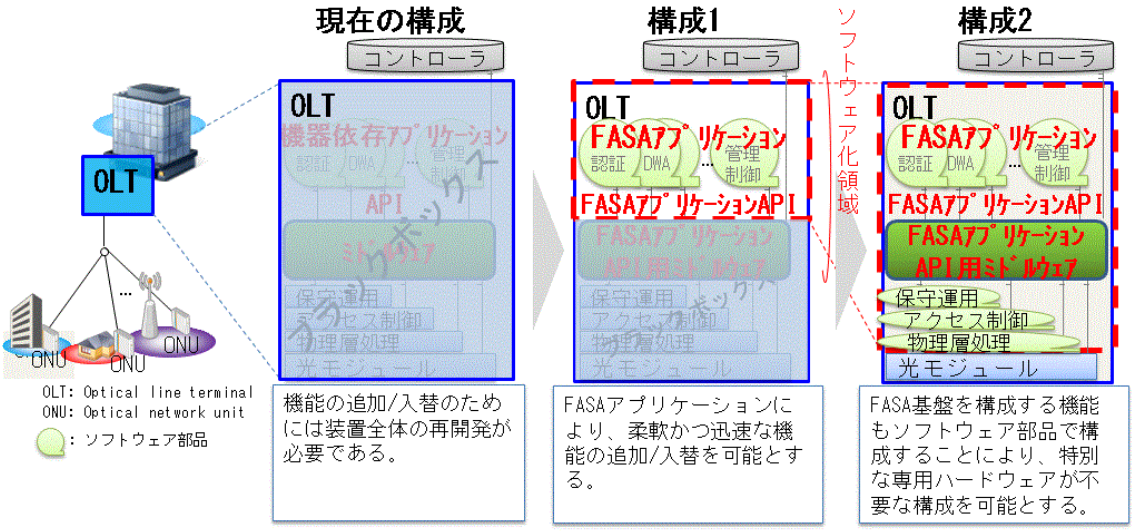


図 1.　　FASA による部品化の構成例

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

1. はじめに

1.1. 本資料の目的

本資料では、様々な通信事業者要件、様々なサービスやアプリケーションへの迅速な対応が可能となる新アクセスシステムアーキテクチャ FASA(Flexible Access System Architecture)と FASA アプリケーションを実装するための汎用化した入出力インタフェースであるFASAアプリケーションAPI(Application Programming Interface)について述べる。

1.2. FASA 概要

図 1.2.-1 に、アクセスシステムの現状とあるべき姿を示す。

現在のシステム・装置構成では、サービス専用のシステム・装置として機能の実装がなされているため、機能の追加や入替にはシステム・装置全体の再開発が必要となる等制約が多く、また、保守運用についてもそれぞれの装置に対応する保守部品やスキルが必要となっている状況である。そのため、図に示すように、アクセスネットワーク装置の柔軟性や拡張性を向上し、様々な通信事業者要件及びサービスへの迅速な対応を可能とすることを目指す必要がある。

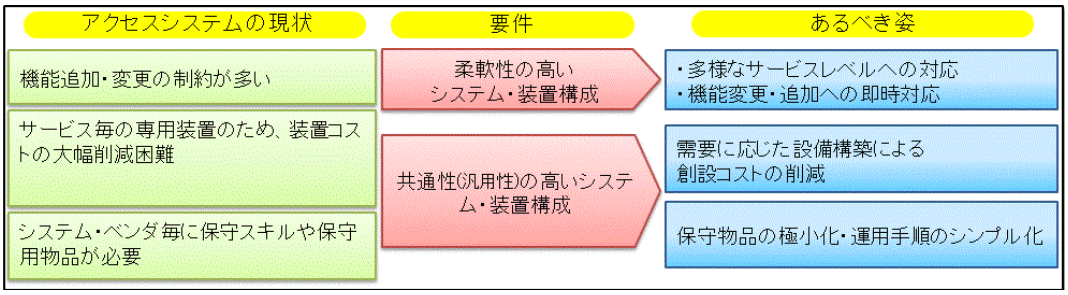


図 1.2.-1　　アクセスシステムの現状とあるべき姿

FASA は、以下を実現する新たなアクセスシステムアーキテクチャ及びそのコンセプトである。

1. これまでのように、機能やサービスに特化した専用装置として開発するのではなく、アクセスネットワーク装置を構成する機能を部品化する。
2. サービス毎及び通信事業者毎に異なる機能は、入出力インタフェースを汎用化したソフトウェア部品で実現する。
3. ソフトウェア部品間の独立性を高めることで、ソフトウェア部品を入替可能な基盤上で動作させることによって、サービス品質を維持しながら、必要な機能をサービス要件に応じて柔軟かつ経済的に実現する。

FASA による部品化の概念図を図 1.2.-2 に示す。図に示すように、FASA に基づくアクセスネットワーク装置は FASA アプリケーションと FASA 基盤とから構成される。

「FASA アプリケーション」は、サービス毎あるいは通信事業者毎に異なる機能を、汎用

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

化した入出力インタフェース(FASA アプリケーション API)を備えたソフトウェア部品で実現し、それらを入替可能としたものである。サービスに応じて、FASA アプリケーションを追加や入替することで、様々な要件のサービスを迅速かつ簡単に提供する。

「FASA 基盤」は、FASA アプリケーションに FASA アプリケーション API を提供するとともに、標準化されているなどの理由で、サービスや要求に応じた変更を行う必要のない機能を提供するアクセス装置の基盤的構成要素である。FASA 基盤では、処理性能等の要件に応じて、FASA 基盤を構成する各機能を、ハードウェアあるいはソフトウェアで実現する。具体的な FASA による部品化の構成例を、図 1.2.-3 に示す。

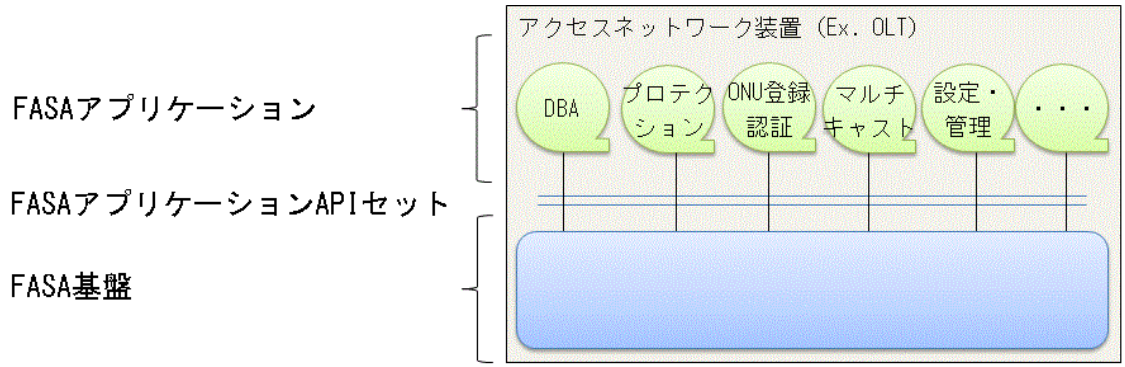


図 1.2.-2　　FASA により実現する部品化の概念図

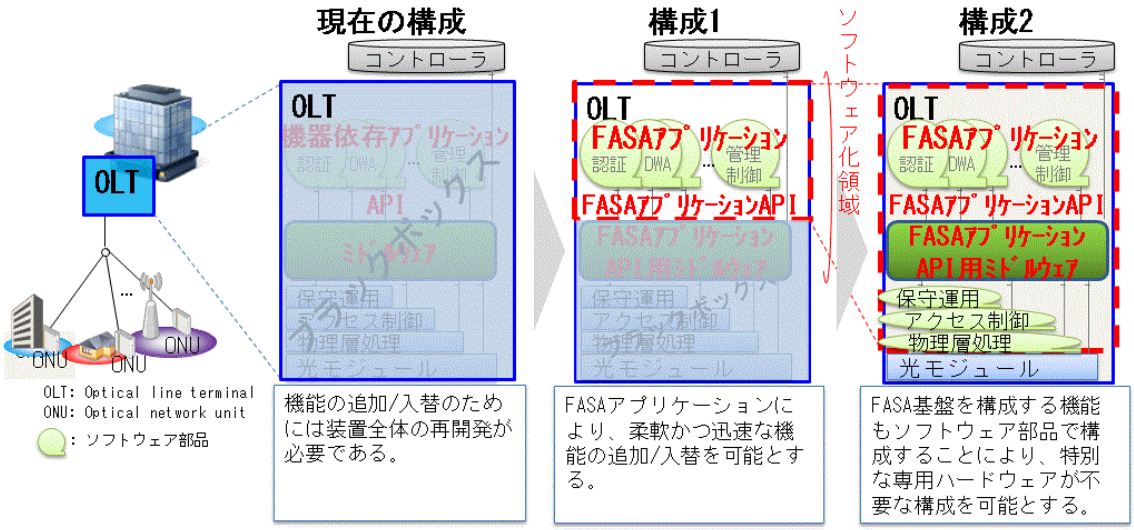


図 1.2.-3　　FASA による部品化の構成例

構成 1 は、FASA アプリケーション API の上部(FASA アプリケーション)をソフトウェア部品化の対象としたものであり、FASA 基盤を構成する各機能の部品化については対象外としている。例えば本構成による NG-PON2 システムの場合、FASA 基盤にて標準化された NG-PON2

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

プロトコルの提供を行い、EPON など他の PON プロトコルへの変更については考慮しない。

構成 2 では、FASA 基盤を構成する各機能についても部品化を行い、ソフトウェア部品化の対象としている。

構成 1 と構成 2 のいずれの構成でも、FASA アプリケーション API は同一である。

FASA 基盤を汎用ハードウェアと外付ハードウェア部品に分割した例を図 1.2.-4 に示す。図に示すアクセスネットワーク装置は、3 つの部品カテゴリ、「(1)FASA アプリケーション」、「(2)汎用ハードウェア」、「(3)外付ハードウェア部品」から構成される。これらを組み合わせることにより必要な機能を迅速かつ簡単に提供する。「(2)汎用ハードウェア」は、アクセスネットワーク装置に限らず汎用的な通信機能を共通部品として実装したものである。汎用ハードウェアによる共通部品化を図ることで、サービス要件に応じて装置を部材レベルから新規に開発する頻度を低減できる。また、共通部品を用いることにより、装置コスト削減や、保守物品種別等の削減による保守運用のシンプル化が進むことが期待できる。「(2)汎用ハードウェア」は、例えば、汎用的なサーバやホワイトボックススイッチ等のアクセスシステム専用装置ではないハードウェアであり、当該ハードウェアに必要なファームウェア、OS 等のソフトウェア、「(1)FASA アプリケーション」のための FASA アプリケーション API 用ミドルウェアを搭載している。「(3)外付ハードウェア部品」は、「(1)FASA アプリケーション」化や「(2)汎用ハードウェア」上への実装が困難な、光送受信部等の機能を、「(2)汎用ハードウェア」と分離して実装したものである。「(3)外付ハードウェア部品」は、例えば光送受信部等の「(2)汎用ハードウェア」以外のハードウェアである。「(3)外付ハードウェア部品」として単純な光モジュールを用いて、メディアコンバータを実現する場合等においては、「(2)汎用ハードウェア」に搭載される OS 等のソフトウェア、「(1)FASA アプリケーション」のための FASA アプリケーション API 用ミドルウェアを用いて動作することとなる。一方、「(3)外付ハードウェア」に PON の MAC チップを搭載する場合等では、当該ハードウェアの動作に必要なファームウェア、OS 等のソフトウェア、「(1)FASA アプリケーション」のための FASA アプリケーション API 用ミドルウェアを「(3)外付ハードウェア」に直接搭載する場合がある。いずれにせよ、サービス要件に応じて例えば伝送容量や伝送方式が異なる外付ハードウェア部品を入替し、それに応じたソフトウェア部品を FASA 基盤に具備すなわち FASA 基盤を構成するソフトウェア部品の入替を行うことで、同一の汎用ハードウェアを用いながら状況に応じた最適な伝送容量や伝送方式のアクセスネットワークの実現が可能である。

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

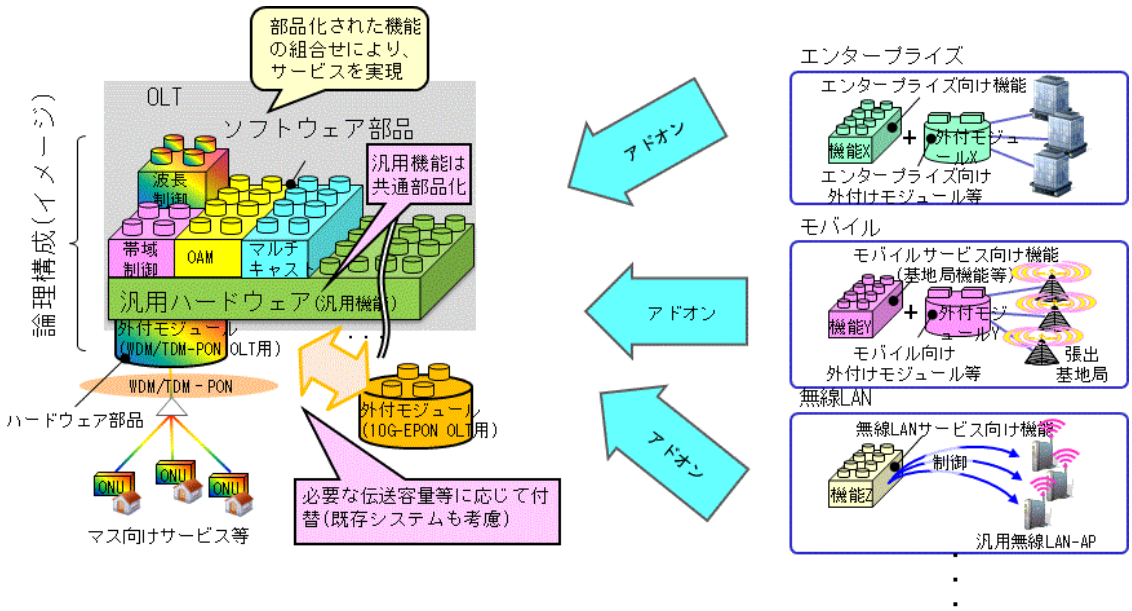


図 1.2.-4　　FASA 基盤の部品化の考え方の例

1.3.　本資料における FASA アプリケーション API の規定範囲

本資料では、FASA を適用したアクセスシステムのユースケースと、ユースケースを実現するための具体的なアーキテクチャ、FASA アプリケーションとする機能及び FASA アプリケーション API について述べる。なお、現時点では対象とするアクセスシステムによっては、処理性能等の観点で PON の MAC チップ等の専用ハードウェアが必要となる場合があることから、FASA 基盤で提供する各機能の部品化(図 1.2.-3 の構成 2)については本資料の対象外とし、図 1.2.-3 の構成 1 の PON OLT を対象とする。

1.4.　用語と定義及び略語

FASA-API: FASA で用いる API の総称。

FASA アプリケーション: FASA アプリケーション API を使って実現された入替可能なソフトウェア部品。

FASA アプリケーション API: FASA アプリケーションと FASA アプリケーション API 用ミドルウェアミドルウェアを接続する API。

FASA アプリケーション API 用ミドルウェア: FASA 基盤のうち、FASA アプリケーションに対して FASA アプリケーション API を提供するソフトウェア。FASA アプリケーション API 用ミドルウェアは、FASA アプリケーション間及びその他の FASA 基盤の機能との通信のための手段を提供するとともに、FASA アプリケーションより下部の機能等の差異を吸収する。

FASA 基盤: FASA アプリケーションに FASA アプリケーション API を提供するとともに、標準化されているなどの理由で、サービスや要求に応じた変更を行う必要のない

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

機能を提供するアクセス装置の基盤的構成要素である。

FASA 基盤 API: FASA 基盤内のソフトウェア部品に用いる API。

外付ハードウェア部品: 汎用ハードウェア上への実装が困難な機能を、汎用ハードウェアと分離して実装したもの。

ソフトウェア部品: 必要な機能を交換可能な単位でソフトウェア化したもの。

汎用ハードウェア: 汎用的なサーバやホワイトボックススイッチのように、アクセスネットワークサービスに限らず汎用的に使えるハードウェア。

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

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

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 変換: Serial to Parallel 変換
SR: Status Reporting
SyncE: Synchronous Ethernet
TBD: To Be Determined
TC: Transmission Convergence
TDD: Time Division Duplex
TDM: Time Division Multiplexing

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

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. 参考文献

[1] ITU-T G.989 シリーズ
[2] IEEE std.802.3TM, 2015
[3] IEEE std.1904.1, 2013
[4] ITU-T G.supplement 51
[5] T. Tashiro, et al., "A Novel DBA Scheme for TDM-PON based Mobile Fronthaul," OFC2014, paper Tu3F.3, Mar. 2014.
[6] D. Hisano, et. al., "Efficient Accommodation of Mobile Fronthaul and Secondary Services in a TDM-PON System with Wireless TDD Frame Monitor, "IEEE ICC 2016, paper ONS1.1, May 2016.
[7] T. Kobayashi, et al., "Bandwidth Allocation scheme based on Simple Statistical Traffic Analysis for TDM-PON based Mobile Fronthaul," OFC 2016, paper W3C.7.
[8] "AT&T Vision Alignment Challenge Technology Survey," AT&T Domain 2.0 Vision White Paper, November 13, 2013.
[9] http://opencord.org/
[10] http://onosproject.org/
[11] https://wiki.opencord.org/display/CORD/VOLTHA:+vOLT+Hardware+Abstraction
[12] Broadband Forum WT-385 "Yang models for management of PON".
[13] IEEE P802.3.2 "Standard for Ethernet YANG Data Model Definitions"

2. 参照アーキテクチャ

2.1. 論理モデル

論理モデルを図 2.1.-1 に示す。モデルは OLT の例であり、コントローラと外部装置は OLT

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

に含まれないが、FASA アプリケーション API との通信を例示するために記載する。論理モデルは、FASA アプリケーションと、FASA アプリケーションに FASA アプリケーション API を提供する FASA 基盤とから構成される。FASA 基盤は FASA アプリケーション用ミドルウェアを含む。

FASA アプリケーション API 用ミドルウェアは、FASA 基盤を構成するハードウェアやソフトウェアのベンダや方式の違いを吸収する。FASA アプリケーション API 用ミドルウェア上にベンダや方式に依存しない FASA アプリケーション API セットを規定し、FASA アプリケーションの入替により、サービス毎あるいは通信事業者毎に必要な機能を実現する。FASA アプリケーション間の通信やコントローラ等による設定管理は FASA アプリケーション API 用ミドルウェアを介して行う。

FASA アプリケーション API セットは、FASA アプリケーションで利用する共通の API 群であり、FASA アプリケーション毎に必要な API を API セットから選択して利用する。

コントローラは、NE-OpS 等の OLT の設定管理システムであり、コントローラからの制御信号は FASA アプリケーション API を介して SBI アプリと通信する。SBI アプリと通信する制御信号は、FASA アプリケーション API を介して制御管理アプリで終端される。

外部装置は、例えばモバイルシステムの BBU や他 OLT であり、FASA アプリケーションと FASA アプリケーション API を介して通信する外部の装置である。図では、外部装置(BBU)が DBA アプリと通信している。

図では、FASA アプリケーション API 用ミドルウェアより下部の機能との通信、アプリ間通信、アプリ間通信の内の設定管理アプリとその他アプリとの通信を、それぞれ赤色、緑色、橙色の矢印で示す。

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

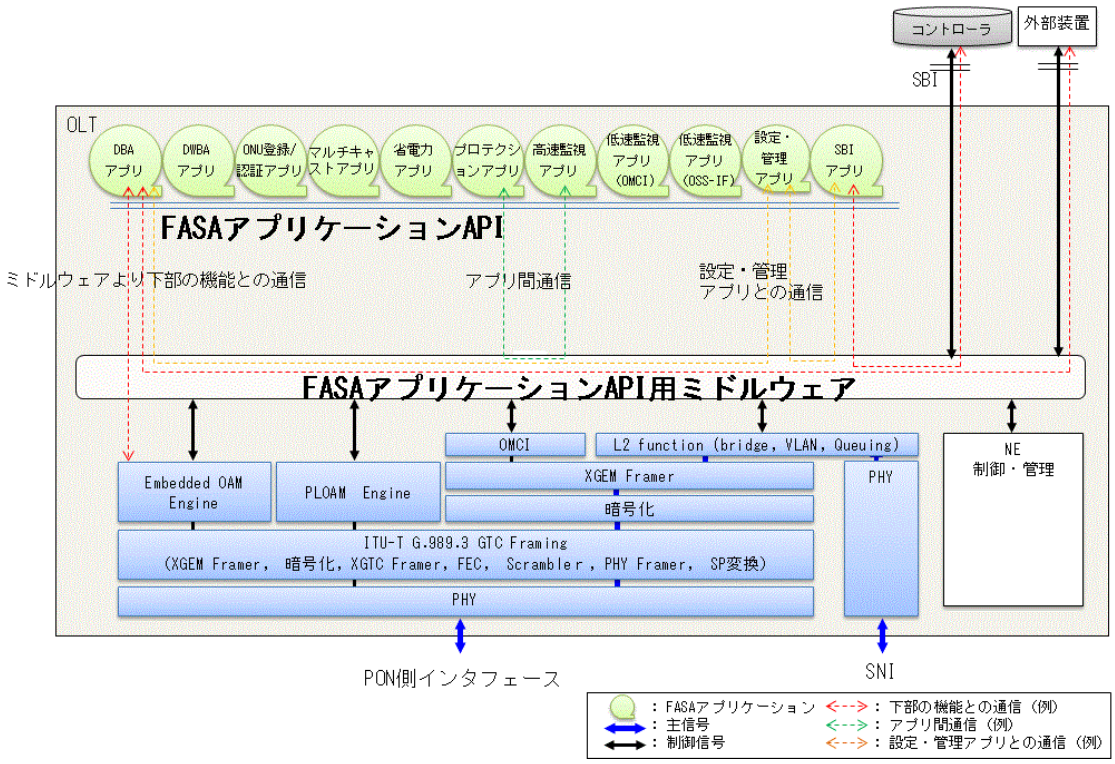


図 2.1.-1　　FASA の論理モデル

2.2.　システム構成例

NG-PON2 OLT を例にとり、FASA に基づくアクセスネットワーク装置の構成例を図 2.2.-1 に示す。図は、FASA 基盤を複数のハードウェア(NG-PON2 ボックス、ホワイトボックススイッチ)で構成した例を示している。NG-PON2 ボックスとホワイトボックススイッチとは、Ethernet 等の標準的なプロトコルで接続した。機能の追加や入替は、FASA 基盤の FASA アプリケーション API 上への FASA アプリケーションの追加や入替により行う。FASA アプリケーション API 用ミドルウェアは、ベンダや方式の差異によるハードウェア及びソフトウェアの違いを吸収し、FASA アプリケーション API を提供するためのソフトウェアである。

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

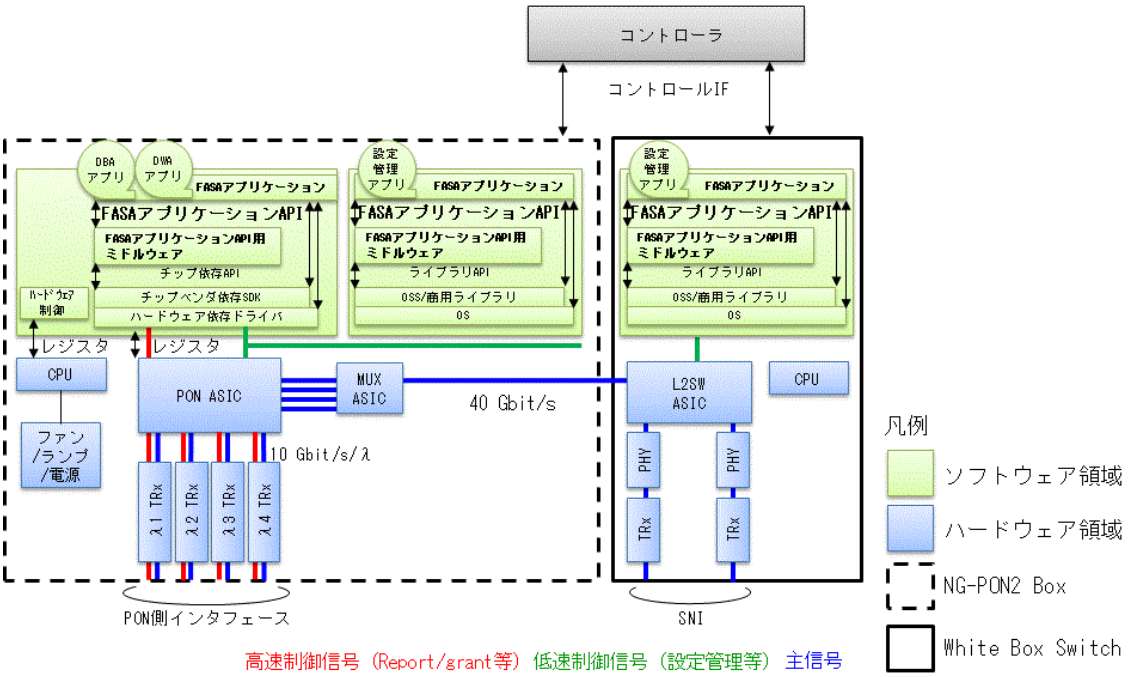


図 2.2.-1　　FASA に基づくアクセスネットワーク装置(NG-PON2 OLT) 構成例

2.3 関連する周辺技術

一方で，FASA と近しいコンセプトとして，AT&T が提唱している CORD (Central Office Re-architectured as a Data center) がある．また，最近ではネットワーク機器の制御方法として，SNMP/MIB に替わるものとして Netconf/YANG の存在感が増している．光アクセス領域における YANG データモデルの標準化の取り組みとして BBF において WT-385/394 の検討も進んでいる．

本節では，CORD 等のこれら周辺技術について述べるとともに FASA との関係について述べる．

2.3.1　CORD

CORD は，AT&T 社が推進する Domain 2.0 [8] における１つのプロジェクトであり，進展著しい SDN および NFV の技術を取り込み，通信事業者の収容局内の装置・システム・ネットワークを最新のデータセンタの如く作り替える，というものである．これにより，OPEX/CAPEX の削減，サービス導入までのリードタイム短縮を狙ったものである [9]．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

2.3.2 ONOS

CORD は，ホワイトボックススイッチなどの OCP(Open Compute project)ハードウェアと，これらを制御するコントローラから構成される．このうち，SDN コントローラとして用いられるのが ONOS(Open Network Operating System)である[10]．ONOS は，通信事業者などから支援を受けている ON. Lab がオープンソースソフトウェアとして開発を行っている．
ONOS から各ハードウェア（通信装置）に対して，Openflow や Netconf/YANG などにより設定・制御・監視を行う．

また，ONOS は OSGi(Apache karaf)をプラットフォームとしており，ONOS 上のアプリケーションは機能追加・更新が容易に可能となっている．

2.3.3 vOLT-HA

vOLT-HA(Virtual OLT Hardware Abstraction)は，CORD プロジェクトの一部である．CORD では，複数ベンダの OLT が準備されているが，これら複数社間のベンダ差分を隠蔽する役割を vOLT-HA が担う[11]．ONOS と同様に，オープンソースソフトウェアとして開発されている．

具体的には，コントローラとしての ONOS と，物理的な通信ハードウェアである OLT との間に位置し，Openflow/Netconf エージェントが ONOS からの設定・制御を受信し，各 OLT に展開する．このとき，vOLT-HA から各 OLT へのインターフェースまでを共通化し，各社の OLT 用のプラグインをそれぞれ備えることで，ONOS あるいは vOLT-HA の中核部からは OLT のベンダ差分を隠蔽する．OLT から ONU へのメッセージ送受信指示なども，引数にメッセージ本体を取ることで，インターフェースとしては共通化されている．

2.3.4 Netconf/YANG

Netconf はネットワーク機器の設定プロトコルであり，YANG はそのデータ管理モデルである．光アクセスにおける標準モデルを作る取り組みが BBF，IEEE であり[12][13]，これらの進展により，EMS/コントローラからの設定・制御がベンダ差分なく共通的なものにできる可能性がある．

2.3.5 FASA との関係

FASA においては，これらのオープン化された技術を積極的に取り込んでいく方針である．中でも vOLT-HA は，現時点では開発中であるものの，ハードウェアに近い領域に位置し，OLT への設定・制御，メッセージ送受信指示などを共通化する取り組みである．今後の開発の推移を見守る必要はあるものの，相応の部分が FASA でも活用できるのではないかと期待される．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

3. ユースケース

FASA のユースケースとして、FASA アプリケーション入替によるマルチサービス収容と、FASA アプリケーション入替による通信事業者独自仕様の実現を示す。

前者は、FASA アプリケーションを入替することで、モバイル、マス、ビジネス等の要求条件が、大きく異なるサービスを PON システムに収容する。3.1.では、その代表的な例である DBA のユースケースについて述べる。

後者は、通信事業者毎に使い方の異なる機能に対応するために、全ての組み合わせを予め実装しておくのではなく、必要に応じて、FASA アプリケーション入替によって所望の機能を備えたアクセスネットワーク装置を提供する。3.2.及び 3.3.では、その代表的な例である省電力化及びプロテクションのユースケースについて述べる。図 3.-1 に、FASA アプリケーション入替による独自仕様の入替イメージを示す。

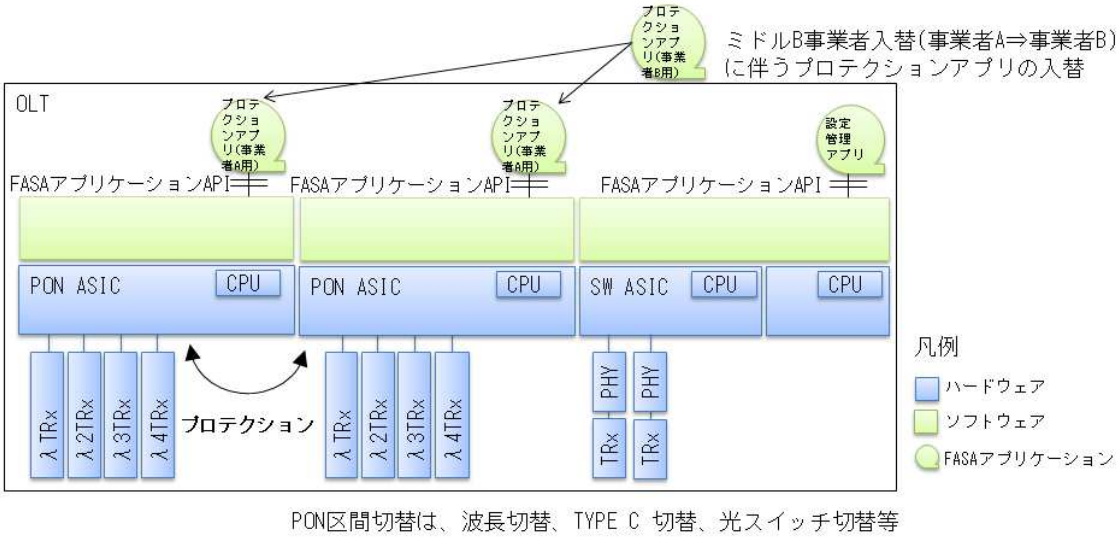


図 3.-1　　FASA アプリケーション入替による独自仕様の入替

3.1. DBA の入替によるマルチサービス収容の実現

現在 FTTH に主に利用されている PON システムを用いて、モバイル、マス、ビジネス等のマルチサービスを効率的に提供することが期待されている。しかし、これらのサービス毎の要求条件は大きく異なっている。特に、モバイルフロントホール(MFH)の主信号に対する最大許容遅延は、例えば従来のインタネットアクセスサービスに比べ厳しく規定されている。そのため、PON の上りの帯域を割当する DBA は、厳しい遅延規定を満たす必要がある。また、MFH への PON の適用に関しては、第五世代モバイル通信などの今後の技術の進展や規格化の進展等により要件が変化することが容易に想定され、その変化にタイムリに追従したシステム開発が必要である。FASA において、FASA アプリケーション入替により、各サー

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

ビスに応じた DBA をタイムリに提供する。

3.1.では、DBA のユースケースの網羅的な分析から、将来のサービス追加や入替に対応可能な API を規定する。

3.1.1. MFH 向け DBA

DBA は、各 ONU の上り帯域(送信可能タイムスロット)を割当する処理であり、ONU からの申告をもとに各 ONU に動的に帯域を割当する SR-DBA と、ONU からの申告を用いない NSR-DBA に分類できる。SR-DBA では、ONU の申告に応じた帯域利用効率の高い割当ができる一方で、OLT-ONU 間の制御信号の往復に時間を要するため、MFH の要件を満たすことが困難である。そこで、外部装置から帯域割当に必要な情報を取得する光モバイル連携 DBA と、トラフィック予測に基づき帯域割当する NSR-DBA を提案している[5-7]。

3.1.1.1. MFH 向け光モバイル連携 DBA

MFH 向け光モバイル連携 DBA の構成とシーケンスを図 3.1.1.1.-1 に示す。MFH 向け光モバイル連携 DBA では、BBU と OLT との連携によって、MFH の遅延規定を満たす帯域割当を実現する。具体的には、基地局 (RRH) からの上りパケット到着に基づく ONU からの帯域割当要求の代わりに、各 UE に対して BBU が生成する上り送信制御情報相当の無線リソース情報を用いて、帯域割当情報(各 ONU の上り帯域とタイミング)を DBA アプリあるいは外部装置で算出して上り帯域の割当を行う[5]。この動作により、上り信号が ONU に到着してから PON 区間に ONU から信号が出力されるまでの待ち時間をなくし、遅延を低減する。本 DBA では、外部装置との連携を可能とする API が必要となる。

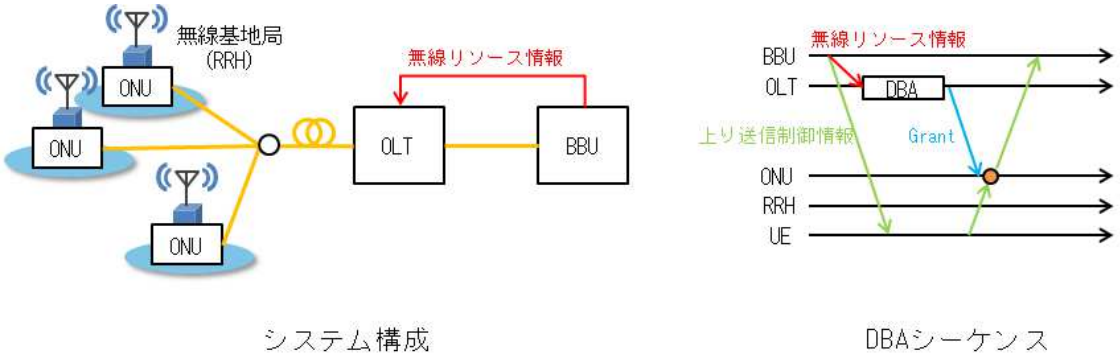


図 3.1.1.1.-1 MFH 向け光モバイル連携 DBA

3.1.1.2. MFH 向け NSR-DBA

図 3.1.1.2.-1 に 2 種類の MFH 向け NSR-DBA の例の概要を示す。これらは、固定帯域割当(FBA)をベースとした NSR-DBA により、SR-DBA で生じる制御信号の往復遅延を削減し、MFH の遅延規定を満たす帯域割当を実現する。具体的には、トラフィックの統計データ及びト

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

ラフィックパターンに基づき NSR-DBA を適用することで、FBA で生じる過剰な割当帯域を削減しつつ、マス向けシステムとの共用等により、システム全体の高効率化を可能にする。

一つ目の NSR-DBA は、TDD 方式のモバイルシステムの TDD 周期をトラフィックの統計データから推定し、TDD 周期に応じて帯域割当を行う[6]。本 DBA では、TDD 周期を推定するために十分なサンプリング速度でトラフィック統計情報を利用可能とする API が必要となる。

二つ目の NSR-DBA は、トラフィックの変動が 1 日～1 週間程度の時間軸で観測すると周期的になっていることに着目し、トラフィック統計情報をもとに次のトラフィック周期で必要な帯域を推定して割当を行う[7]。本 DBA では、長期に渡るトラフィック統計データを利用可能とする API が必要となる。

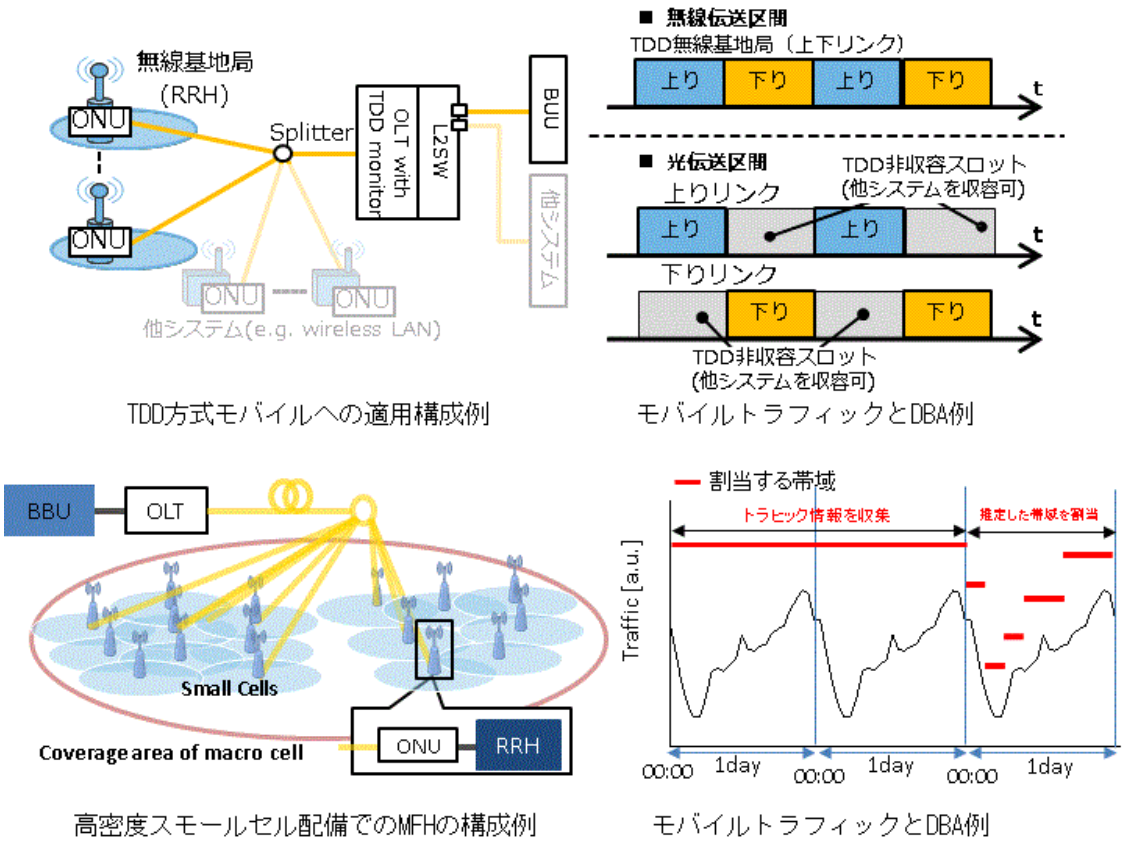


図 3.1.1.2.-1　　MFH 向け NSR-DBA

3.1.2.　マス向け SR-DBA

図 3.1.2.-1 に、マス向け SR-DBA の構成とシーケンスを示す。マス向け SR-DBA では、各 ONU の申告から抽出した要求量をもとに、各 ONU に動的に帯域割当を行う。よって、本 DBA では、各 ONU の申告から要求量を抽出する API が必要である。

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

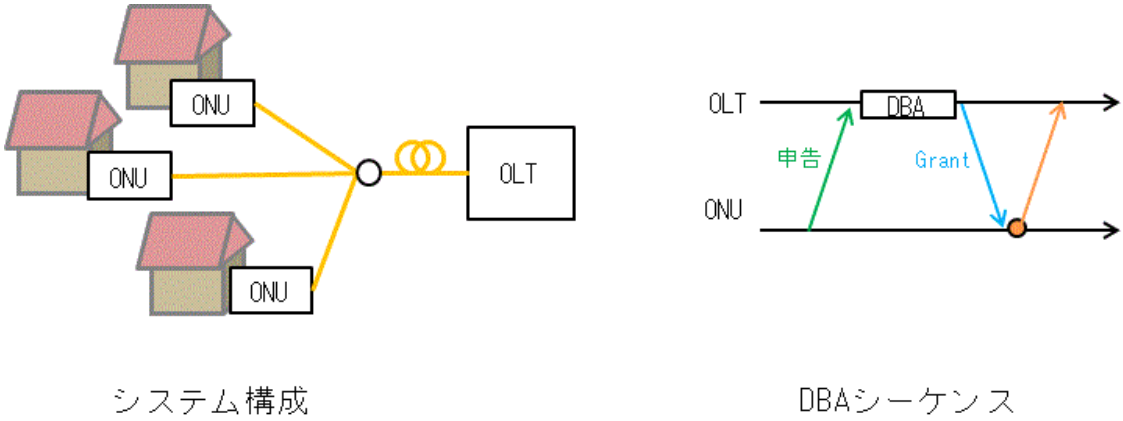


図 3.1.2.-1　　マス向け SR-DBA

3.1.3.　ビジネス向け DBA

TBD

3.1.4.　DBA の API と機能ブロック

以上から、各ユースケースを実現するための DBA の API と機能ブロック図を図 3.1.4.-1 に示す。また、図 3.1.4.-1 の各機能部(方針決定部、割当計算部、連携制御部、トラフィックモニタ、申告処理部、grant 処理部)の定義を、図 3.1.4.-2 に示す。

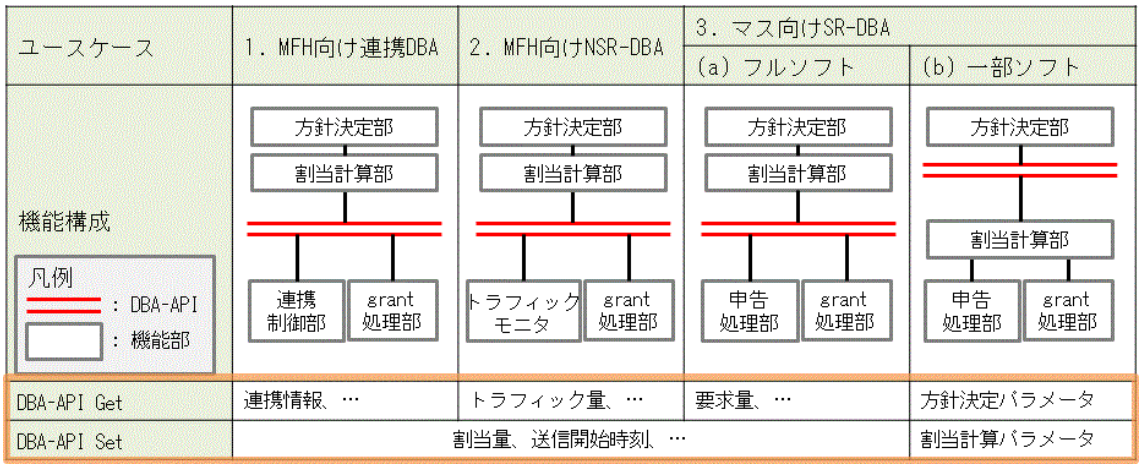


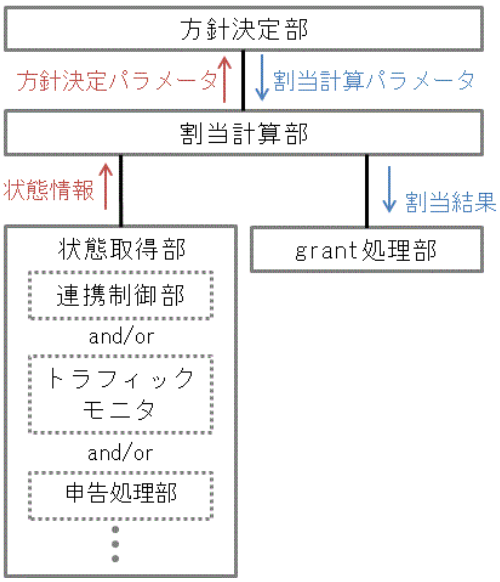
図 3.1.4.-1　　DBA の API と機能ブロック図

図 3.1.4.-1 のユースケース 1 及び 2 は、外部装置の連携等の DBA を柔軟に実装するために、DBA 全体(方針決定部及び割当計算部)を FASA アプリケーションとして実装する場合の構成と API の位置を示す。また図 3.1.4.-1 のユースケース 3 のマス向け SR-DBA は、DBA 全体を FASA アプリケーションとして実装する「(a)DBA 機能のフルソフト化」に加え、従来の

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

PON の MAC チップ等と同様な構成として DBA の一部(方式決定部)のみを実装する「(b)DBA 機能の一部ソフト化」の場合の構成と API の位置を示す。

以上のユースケースに対応するために、FASA 基盤は、PON の標準で規定されている申告処理機能、grant 処理機能に加えて、連携制御機能、トラフィックモニタ機能を有し、連携情報取得、トラフィック量取得、要求量取得、割当計算パラメータ取得、割当量設定、送信開始時刻設定、方針決定パラメータ設定の API を提供する必要がある。



機能部	概要
方針決定部	任意のタイミングで、割当計算部から取得した方針決定パラメータに基づき、割当計算パラメータを計算し、割当計算部に設定する機能部。
割当計算部	帯域割当毎に、状態取得部から取得した状態情報と、方針決定部から設定された割当計算パラメータに基づき、帯域割当計算し、割当結果をgrant処理部に設定する機能部。
状態取得部	割当対象の状態情報を取得する機能部。例えば、連携制御部、トラフィックモニタ、申告処理部などがある。 • 連携制御部：モバイルシステムから無線リソース情報を取得する機能部。無線リソース情報から帯域割当情報を算出する場合もある。 • トラフィックモニタ：特定の期間、特定のフローのトラヒック量を計測する機能部。 • 申告処理部：割当対象から申告を受信し、要求量を抽出する機能部。例えばNG-PON2においては、上りフレームからBufOccを取得する機能部に相当する。
grant処理部	割当計算部から設定された割当結果をもとにgrantを生成、送信する機能部。例えばNG-PON2においては、BWmapを作成・送信する機能部に相当する。

図 3.1.4.-2　　機能部の定義

3.2.1. 省電力機能の入替による通信事業者独自仕様の実現

TBD

3.2.2. プロテクション機能の入替による通信事業者独自仕様の実現

通信事業者要件として、アクセスシステムに対しても高い可用性が必要となる。装置故障時の対応のみならず、アクセスネットワーク装置のソフトウェア更新や処理性能が向上した後継ハードウェアへ移行する際も、サービス停止なく移行する必要があるため、プロテクション機能が重要となる。

通信事業者毎に、アクセスネットワーク装置と接続するネットワークの構成が異なることから、異なる方式や異なるポリシーでプロテクションを行うことが想定されるため、FASA ではプロテクション用 API を規定する。

4. アクセスシステムの主要機能と FASA アプリケーション化の対象

アクセスシステムの主要機能と FASA アプリケーション化の対象を表 4.-1 に示す。まず、

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

アクセスシステムの主要機能について説明する。

PON 主信号処理機能は、ONU との間で送受信を行う主信号を処理する機能群であり、PON フレームの生成分離や前方誤り訂正(FEC: Forward Error Correction)等の機能を含む。

PON アクセス制御機能は前述の主信号送受信を行うための制御機能群であり、動的帯域割当や ONU の登録認証等の機能を含む。

L2 主信号処理機能は、PON 側ポートと SNI 側ポートとの間で主信号を転送し、処理する機能群であり、MAC アドレス学習や VLAN 制御、優先制御やトラフィックモニタ等の機能を含む。

保守運用機能は、アクセス装置によってサービスを円滑に保守運用するための機能群であり、ONU や OLT(OSU 及びスイッチ)の装置設定を実施する機能や、ソフトウェアの更新、装置やサービスの管理、各種機能が正常に動作しているかを監視する機能、異常発生時に能動的に警報を発出する機能、異常発生時の範囲や原因を調査するための試験機能等を含む。また、保守運用機能は、多数のアクセス装置を管理する保守運用システムと接続され、リモートからも円滑な保守運用を実現する。

PON マルチキャスト機能は、SNI 側から受信したマルチキャストストリームを適切なユーザに転送する機能群であり、マルチキャストストリームの識別や振分、ONU のフィルタ設定を実施する機能を含む。

省電力制御機能は、ONU や OLT の電力消費を削減するための機能群であり、標準化で規定されている省電力化機能に加え、トラフィックモニタとの連携によってサービスへの影響を最小限に抑えながら、最大限の省電力効果を得るための機能を含む。

周波数/時刻同期機能は、ONU 配下の装置に正確な周波数同期や時刻同期を提供するための機能群であり、自身のリアルタイムクロックを上位装置に従属同期させる機能や、PON フレームを用いて ONU に時刻情報を通知する機能を含む。

プロテクション機能は、スイッチ間や OSU 間等、複数のハードウェアで冗長をとった構成において、障害検知時に現用系から予備系への切替や引継を実施してサービスを継続するための機能群であり、切替トリガの検出や切替処理の実施といった機能を含む。また、プロテクション機能は障害検知時や手動での切替時に、サービスを全面停止せず縮退運転で動作させ続けるための機能を提供する。

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

表 4.-1　　アクセスシステムの主要機能と FASA アプリケーション化の対象

機能群	機能	FASAアプリケーションAPI利用による拡張例	FASAアプリケーション/FASA基盤
PON主信号処理機能	基本機能	標準準拠のフレーム/ヘッダの生成・分離、フラグメント処理、スクランブラ、FEC、暗号化。	基盤
PONアクセス制御機能	ONU登録/認証	独自認証手順(不正利用抑止)	アプリ
	DBA	標準準拠のDBAフレーム処理(フレームヘッダの読み書き等)	基盤
		独自QoS規定(遅延、公平性、帯域利用効率等)への対応	アプリ
		サービス要求(例: モバイル提供のための低遅延要求)への対応	アプリ
		プログラムの暗号化への対応	アプリ
	DWA	標準準拠の波長切り替えシーケンス(ITU-T G.989.3)	基盤
		独自QoS規定(遅延、公平性等)への対応	アプリ
		独自保守運用システムとの整合性確保	アプリ
		トラフィックモニタとの連動による特性向上	アプリ
	DoS対策	独自フィルタ(特定のフィールドとのマッチ、かつ一定以上の流量等)の適用	アプリ
L2主信号処理機能	SNIポート	ミラーリングポートの具備、マルチキャストポートの具備、コネクタ種別、送受信フレーム(Ethernet)、オートネゴシエーション、伝送媒体の指定	基盤
	ブリッジ機能	標準準拠のブリッジ処理(MACアドレス学習、IEEE 802.1Q VLAN tag処理等)	基盤
	トラフィックモニタ	トラフィックモニタの具備(具体的な項目は仕様化が必要)	基盤
	集線	ポリシング機能の具備、メータ/マーカの具備、	基盤
		独自QoS規定(遅延、パケットロス、公平性等)への対応	アプリ
	優先制御	標準準拠のキュー/スケジューラ(SPQ)の具備	基盤
		独自QoS規定(遅延、パケットロス、公平性等)への対応	アプリ
	VLAN管理	独自保守運用システム(PON区間のVLANの使い方等)との整合性確保	アプリ
保守運用機能	保守運用フレーム処理	標準準拠の保守運用フレーム（OMCI, PLOAM, OAM）処理	基盤
	保守運用ポート/監視制御ポート	コネクタ種別、送受信フレーム(Ethernet)、オートネゴシエーション、伝送媒体の指定	基盤
	SBI	コントローラとの制御信号のやりとり、ONUやOLTが他の伝送装置やサーバと連携を取るためのタイミング制御機能	アプリ
	設定	CLIの具備	基盤
		FASAアプリケーション追加、変更時への対応(更新手段の提供等)	アプリ
		独自保守運用システムとの整合性(保守運用システムとの接続プロトコル、交換パラメータ等。例: ユーザの契約サービスの取得)確保	アプリ
	管理	独自ソフトウェア更新方式(転送方式、更新タイミング、更新処理時間、断時間の許容値等)への対応	アプリ
		独自保守運用システムとの整合性（ONUの設定等)確保	アプリ
		独自ログ保存手段(保存エントリ数、形式等)	アプリ
	監視制御	SNMPエージェント具備	基盤
		Ether-OAM具備	基盤
		独自保守運用システムとの整合性確保(警報通知ポートの設定、警報通知用プロトコル等)	アプリ
		独自の監視項目、独自の警報発出条件への対応	アプリ
	試験	独自試験(往復遅延時間測定)への対応等	アプリ
PONマルチキャスト機能	IPマルチキャスト	マルチキャストフレーム（IGMPやMLD）処理	基盤
	フィルタ設定	独自フィルタ(VLAN/MACアドレスを用いる方式等)の適用	アプリ
	マルチキャストプロキシ	独自QoS規定(Zapping性能、信頼性等)	アプリ
		独自サービス(映像サービス等)との整合性確保等	アプリ
省電力制御機能	ONU省電力化	標準準拠のシーケンス(ITU-T G.989.3やIEEE 1904.1に)準拠	基盤
		独自QoS規定(遅延時間等)への対応	アプリ
	OLT省電力化	OLT省電力化に必要な状態遷移の具備	基盤
		独自QoS規定(遅延時間等)への対応	アプリ
周波数/時刻同期機能	同期手段	標準準拠(IEEE 1588等)の同期方式	基盤
		独自サービスとの整合性確保	アプリ
プロテクション機能	プロテクション	プロテクションに必要な状態遷移の具備(仕様化は必要)	基盤
		独自QoS規定(断時間等)への対応	アプリ
		複数の冗長構成(SW冗長、OSU冗長、1+1冗長、N:1冗長等)への対応	アプリ

次に、各機能を FASA アプリケーションとして実装するか、あるいは FASA 基盤上で実装するかの考え方と例について説明する。

表 4.-1 に示した機能のうち、サービスによって機能変更が必要になるものや通信事業者

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

独自の要件を満たすために拡張すべき機能をFASAアプリケーションとして実現する。一方、標準化等で規定されているため拡張の余地が少ない機能はFASA基盤上に実装される。

表4.-1では、例えば、PON主信号処理機能をFASA基盤として実現することを示している。ITU-T G.989シリーズに準拠した40 Gbit/s級のアクセス装置を実現するには、フレームフォーマットや、フレームの暗号化、前方誤り訂正（FEC）機能といった基本的なPON主信号処理機能は標準に従って実装する必要がある。また、こうした基本機能はサービスによらず共通であるため、FASA基盤上に実装される。

別の例として、表4.-1では、PONアクセス制御機能に含まれるDBA機能の「サービス要求への対応」をFASAアプリケーションとして実現することを示している。3章で述べたように、提供するサービスによって、低遅延性を提供するケースや効率良く多数のユーザに帯域を割り当てるケースが存在する。サービス毎に異なる要求を満たすため、帯域割当の手順やポリシーを FASA アプリケーションとして、標準的な処理(標準で規定されている、BWmapフォーマットへの変換等)からは分離することが望ましい。

また、提供するサービスの対象が同じマス向けであっても、通信事業者によってヘビーユーザへの対応方針が異なる等、公平性のポリシーが異なることが考えられる。具体的には、例えばPON単位といった粒度の小さい公平制御を必要とする通信事業者はDBAのアプリケーション内部でも公平制御を行い、アクセス装置単位といった大きい粒度でのみ公平制御を行う通信事業者は集線機能を用いることで、それぞれの QoS 規定を満たすことを想定している。

このように、FASAでは異なる要求をFASAアプリケーションの入替によって実現するため、FASA アプリケーション入替の手段が必要となるが、入替手段として何を採用するかは、通信事業者や運用によって異なる。例えば、通信事業者が使用している既存の保守運用システムがソフトウェア更新にTFTP(Trivial File Transfer Protocol)を用いる場合はTFTPを備え、保守運用システムの外部から SFTP(SSH File Transfer Protocol)を用いて更新する場合はSFTPを備える。また、今後、装置とコントローラ間のインタフェースに関して標準化の議論が進展すると想定しており、標準化の進展に追従したインタフェースの追加や変更についても考慮する必要がある。このため、表4.-1ではアクセス装置が接続する他システムやその運用に合わせてカスタマイズが必要となる機能もFASAアプリケーションとして実現する。

また、FASAでは、FASA基盤全体を完全二重化して行うプロテクションに限らず、FASA基盤の一部のみで行うプロテクションについても想定する。例えばFASA基盤が、光スイッチを備えて PON プロテクションに対応する場合や、一つの PON に対して複数波長を備えて波長プロテクションに対応する場合、スイッチのみを二重化する場合、あるいはこれらを組み合わせた場合等、複数の冗長構成が考えられる。プロテクション機能をFASAアプリケーションとして実装することで、期待する冗長構成に対応でき、また該当箇所を再利用することで、容易に多様な冗長構成にも対応できる。

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

5. API 規定

API は、追加や入替する FASA アプリケーションによらずに、共通的な API セットとし、共通的な API セットから各 FASA アプリケーションで利用する API を選択する。以降では、ITU-T 勧告に基づく PON システムを中心に記述するが、IEEE 標準の PON や PON 以外のアクセスシステムについても対象である。

なお，以下の記述については，現時点で確定したものではなく，今後の検討の進展，あるいは周辺技術の発展に合わせて，適宜更新される．

5.0.1 API の考え方

FASA アプリケーションから下側処理部を見ると，そのインターフェースである API は３種類に大別できる．１つは，OLT 自身に対する設定・制御，情報取得であり，もう１つは，ONU とのメッセージ送受信である．残りの１つは，これら２つのいずれにも当てはまらないものである．

5.0.2 設定・制御 API

API 上部側，すなわち，FASA アプリケーション側の役割としては，コントローラ/EMS からの設定指示・制御メッセージを Netconf/YANG 等で受け取り，その内容に従い，API 下部側処理部に指示をすることが挙げられる．あるいは，OLT 自身の情報取得，情報通知をコントローラ/EMS に転送することもある．

このとき，基本的には YANG モデル等に基づき，メッセージを展開することとなる．これは，vOLT-HA の役割の１つとほぼ同じであり，ここでの設定・制御 API とは，vOLT-HA での OLT 向けインターフェースと同じ役割を担うことになる．そのため，vOLT-HA インターフェースを活用できる可能性が十分にある．さらには，今後の標準 YANG モデル化が十分に進めば，vOLT-HA インターフェースを用いるまでもなく，標準的な設定・制御インターフェースが準備されることも期待できる．

5.0.3 ONU とのメッセージ送受信 API

また，設定・制御を ONU に対して行う場合，あるいは，何らかの指示・情報取得を ONU に対して行う場合，API 上部側の役割としては，ONU に向けたメッセージを組み立て，API 下部側に渡し，送信指示を行う，あるいは，API 下部側からのメッセージを読み出す，ことになる．ONU とのメッセージ交換には，拡張 OAM や OMCI など複数のプロトコルがあるものの，インターフェースとしてみれば，メッセージの送信指示，読み出しに集約できる．これも，vOLT-HA インターフェースを活用できる可能性が十分にある．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

5.0.4　その他

例外的には，OLT ではない他の機器と連携する場合には，そのインターフェースが必要である．

5.0.5　時間制約のある処理

上に記したように，API 上部側の FASA アプリケーションは複数の種類がありうると言えども，API 下部側へのインターフェースとしては，数種類に大別される．さらに，今後の vOLT-HA インターフェース，あるいは標準的なデータモデルに基づくインターフェースなどを活用できる可能性が大きいと想定している．

しかしながら，上記は基本的に時間制約が無いか緩やかな処理であることを前提としている．ここで，API 上部側の時間制約の厳しい機能を FASA アプリケーションとして，モジュール化したいと考えるとすると，上記では不十分である．そのような機能は多くは無いものの，ONU との高頻度なメッセージングを必要とする DBA 機能と，ONU スリープ機能とが考えられる．

これらの機能の内部ルーチンである DBA アルゴリズムもしくは ONU スリープアルゴリズムを独自のものにしたい，すなわち API 上部側に配置するためには，vOLT-HA のインターフェース，あるいは標準データモデルのインターフェースでは出来ず，専用の API が必要である．本資料では，5.2.1 節において，それらについて記す．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

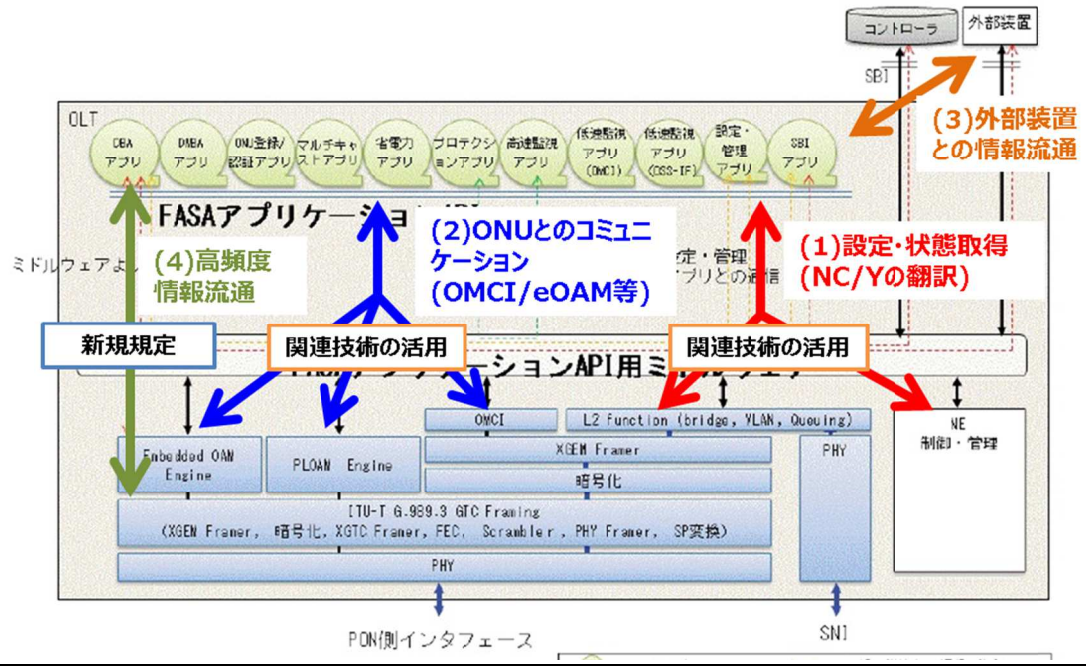


図 5.0.5-1 インターフェースの大別

5.1. PON 主信号処理機能

TBD

5.2. PON アクセス制御機能

5.2.1. DBA

表 5.2.1.-1 に、DBA の各ユースケース(MFH 向け光モバイル連携 DBA、MFH 向け NSR-DBA、マス向け SR-DBA(DBA 機能のフルソフト化)、マス向け SR-DBA(DBA 機能の一部ソフト化))の API リストを示す。表の「ユースケース」列は、ユースケースと API の対応を示す。「ユースケース」の「1」「2」「3a」「3b」の各列にチェックが入った API が、それぞれ MFH 向け光モバイル連携 DBA、MFH 向け NSR-DBA、マス向け SR-DBA(DBA 機能のフルソフト化)、マス向け SR-DBA(DBA 機能の一部ソフト化)で利用される。なお、本資料では、イベント駆動型の DBA プログラムを想定する。

API は、追加や入替する DBA の FASA アプリケーション(DBA アプリ)によらずに共通的な API セットとし、共通的な API セットから各 DBA アプリで利用する API を選択する。例えば、「get_AllocationDbaConfig」「set_GrantSize」「set_GrantSizeAndStartTime」は DBA アプリで共通に利用し、「get_CooperationConfig」「get_DataSize」「get_Traffic」「get_AllocUnitTraffic」「get_CalcConfig」は特定の DBA アプリが利用する。共通に利用する「get_AllocationDbaConfig」は、応答データから、それぞれの DBA アプリで必要な情

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

報を取得する。送信開始時刻を指定しない「set_GrantSize」は FASA 基盤にて、送信開始時刻を指定する「set_GrantSizeAndStartTime」は FASA アプリケーションにて、それぞれ送信時間開始時刻を決定する。

表の「割当対象」は、DBA で帯域割当の対象となる論理パスを示す。割当対象番号は、割当対象のインデックスである。コンフィグ番号は、設定パラメータのインデックスである。モニタ番号は、トラフィックモニタの計測対象のインデックスである。方針決定用パラメータは、方針決定部の計算で用いられるパラメータである。割当計算用パラメータは、割当計算部の計算で用いられるパラメータである。なお、表中の＜詳細未定＞と記載した項目、インデックスと設定パラメータの対応、方針決定用パラメータの内容及び割当計算用パラメータの内容は今後の検討事項である。

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

表 5.2.1.-1　　DBA の API

API	入力	出力	概要	ユースケース			
				1	2	3a	3b
on_Init	(なし)	(なし)	初期化時に呼出されるコールバック関数。	✔	✔	✔	✔
on_Process	(なし)	(なし)	周期的に、あるいは特定の条件を満たした際に(不定期的に)呼出されるコールバック関数。	✔	✔	✔	✔
on_ConfigUpdated	(なし)	<詳細未定>更新された設定情報	設定情報が更新された際に呼出されるコールバック関数。	✔	✔	✔	✔
on_Killed	(なし)	(なし)	DBAプログラムがKillされた際に呼出されるコールバック関数。(例えば、DBAアプリの入替時の利用を想定)	✔	✔	✔	✔
get_DbaConfig	コンフィグ番号	<詳細未定>設定値	DBAコンフィグ情報を取得する関数。(例えば、DBA周期、Quiet Window Size等を想定)	✔	✔	✔	✔
get_AllocUnitDbaConfig	割当対象番号、コンフィグ番号	<詳細未定>設定値	当該割当対象のDBAコンフィグ情報を取得する関数。(例えば最大帯域、保証帯域等を想定)	✔	✔	✔	✔
get_CooperationConfig	コンフィグ番号	<詳細未定>設定値	連携制御部のコンフィグ情報を取得する関数。(例えば連携制御部の動作周期や動作モード、連携先のシステム名、バージョン等を想定)	✔			
get_TrafficMonitorConfig	コンフィグ番号	<詳細未定>設定値	トラフィックモニタのコンフィグ情報を取得する関数。(例えばトラフィックモニタの動作モード等を想定)		✔		
get_CalcConfig	コンフィグ番号	<詳細未定>設定値	割当計算部のコンフィグ情報を取得する関数。(例えば割当計算部の動作モード等を想定)				✔
get_ReportConfig	コンフィグ番号	<詳細未定>設定値	申告処理部のコンフィグ情報を取得する関数。(例えば申告処理部の動作モード等を想定)			✔	✔
get_GrantConfig	コンフィグ番号	<詳細未定>設定値	grant処理部のコンフィグ情報を取得する関数。(例えばgrant処理部の動作モード等を想定)	✔	✔	✔	✔
set_DbaConfig	コンフィグ番号、<詳細未定>設定値	(なし)	DBAコンフィグ情報を設定する関数(例えばDBA周期、Quiet Window Size等を想定)	✔	✔	✔	✔
set_AllocUnitDbaConfig	割当対象番号、コンフィグ番号、<詳細未定> 設定値	(なし)	当該割当対象のDBAコンフィグ情報を設定する関数。(例えば最大帯域、保証帯域等を想定)	✔	✔	✔	✔
set_CooperationConfig	コンフィグ番号、<詳細未定>設定値	(なし)	連携制御部のコンフィグ情報を設定する関数。(例えば連携制御部の動作周期や動作モード、連携先のシステム名、バージョン等を想定)	✔			
set_TrafficMonitorConfig	コンフィグ番号、<詳細未定>設定値	(なし)	トラフィックモニタのコンフィグ情報を設定する関数。(例えばトラフィックモニタの動作モード等を想定)		✔		
set_CalcConfig	コンフィグ番号、<詳細未定>設定値	(なし)	割当計算部のコンフィグ情報を設定する関数。(例えば割当計算部の動作モード等を想定)				✔
set_ReportConfig	コンフィグ番号、<詳細未定>設定値	(なし)	申告処理部のコンフィグ情報を設定する関数。(例えば申告処理部の動作モード等を想定)			✔	✔
set_GrantConfig	コンフィグ番号、<詳細未定>設定値	(なし)	grant処理部のコンフィグ情報を設定する関数。(例えばgrant処理部の動作モード等を想定)	✔	✔	✔	✔
get_DataSize	時刻、割当対象番号	要求量	当該割当対象の要求量を連携制御部から取得する関数。	✔			
get_TimeAndDataSize	割当対象番号	時刻、要求量	当該割当対象のONU到着時刻と要求量を連携制御部から取得する関数。	✔			
get_Traffic	モニタ番号	<詳細未定> モニタ結果	当該モニタ番号のモニタ結果を取得する関数。(割当対象に紐付かないモニタ結果取得API)		✔		
get_AllocUnitTraffic	割当対象番号、モニタ番号	<詳細未定> モニタ結果	当該割当対象の当該モニタ結果を取得する関数。		✔		
get_ReportSize	割当対象番号	要求量	当該割当対象の要求量を取得する関数。			✔	
get_ParameterForPolicyMaking	方針決定用パラメータ番号	<詳細未定> 方針決定用パラメータ値	当該方針決定用パラメータ番号の値を取得する関数。(例えば、直近の要求量の総和や、直近の割当量の総和等を想定)				✔
get_AllocUnitParameterForPolicyMaking	割当対象番号、方針決定用パラメータ番号	<詳細未定> 方針決定用パラメータ値	当該割当対象の当該方針決定用パラメータ番号の値を取得する関数。(例えば、当該割当対象の過去の要求量及び割当量の履歴等を想定)				✔
set_GrantSize	割当対象番号、割当量	(なし)	当該割当対象に対する割当量をgrant末尾に追加する関数。(送信開始時刻はgrant処理部が自動計算)	✔	✔	✔	
set_GrantSizeAndStartTime	割当対象番号、割当量、送信開始時刻	(なし)	当該割当対象に対する割当量及び送信開始時刻をgrantに追加する関数。	✔	✔	✔	
set_ParameterForCalc	割当計算用パラメータ番号、<詳細未定> 割当計算用パラメータ値	(なし)	当該割当計算用パラメータ番号の値を設定する関数。(例えば、全割当対象に共通の追加割当量等を想定)				✔
set_AllocUnitParameterForCalc	割当対象番号、割当計算用パラメータ番号、<詳細未定>割当計算用パラメータ値	(なし)	当該割当対象の当該割当計算用パラメータ番号の値を設定する関数。(例えば、当該割当対象への追加割当量等を想定)				✔

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

5.2.1.1. MFH 向け光モバイル連携 DBA

表 5.2.1.-1 でユースケース 1 の欄にチェックのある API が､MFH 向け光モバイル連携 DBA の利用する API の例である。例えば「get_CooperationConfig」は、連携制御部の動作周期や動作モード、連携先のシステム名、バージョン等を取得する。「get_DataSize」は、帯域割当毎に呼出される。「get_DataSize」は、個別あるいは指定範囲の割当対象をまとめて取得する。

5.2.1.2. MFH 向け NSR-DBA

表 5.2.1.-1 でユースケース 2 の欄にチェックのある API が、MFH 向け NSR-DBA の利用する API の例である。例えば「get_Traffic」「get_AllocUnitTraffic」は帯域割当毎に呼出される。「get_Traffic」は複数 OLT あるいは複数 OSU あるいは複数 CT のフローを、「get_AllocUnitTraffic」は割当対象の全体あるいは割当対象の特定のフローを、それぞれ主に計測対象とする。

5.2.1.3. マス向け SR-DBA(DBA 機能のフルソフト化)

表 5.2.1-1 でユースケース 3a の欄にチェックのある API がマス向け SR-DBA(DBA 機能のフルソフト化)で利用する API の例である。例えば「get_ReportSize」は帯域割当毎に呼出される。「get_ReportSize」は、個別あるいは指定範囲の割当対象の値を取得対象とする。

5.2.1.4. マス向け SR-DBA(DBA 機能の一部ソフト化)

表 5.2.1.-1 でユースケース 3b の欄にチェックのある API がマス向け SR-DBA(DBA 機能の一部ソフト化)で利用する API の例である。API「get_CalcConfig」の「割当計算部のコンフィグ情報」は、例えば割当対象毎の最大帯域、保証帯域等の設定情報である。「方針決定用パラメータ」は要求量や割当量の履歴等である。

5.2.1.5 DBA アルゴリズムとメッセージングの分離

また，上記とは別の考え方として，API 上側が DBA アルゴリズムの実行を担い，下側の処理部がメッセージングを担う，という構成もある．これは，メッセージング自体は共通なものを用いるが，アルゴリズムは独自のものを搭載したい場合などに有効である．このとき，アルゴリズムに特に依存せずにインターフェースを規定することで，より汎用化，多くの場合に有効になりうる．

このとき必要になるインターフェースとしては，(1)上側(アプリケーション)から上り送信許可に関する全情報を下側（処理部）に通知するもの，(2)下側（処理部）から上り送信要求に関する全情報を上側（アプリケーション）に通知するもの，が必要である．

これら２つのインターフェースにより引き渡される情報は，渡された先での再計算を必要

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

としない値であることが望ましい．これは，上部側と下部側での依存性を減らすためであり，独立性を高めることで，上部側は DBA アルゴリズムのみについて，下部側はメッセージ実装についてのみを考えることができるようになる．

5.2.1.5.1　送信許可量設定 API

形式：

fasa_api_set_grant_config(UINT64 sfc, UINT8 ch, int n_of_configs, grant_config_t grant_config[]);

引数：

```
  UINT64 sfc;  /* superframe カウンタ値 IEEE802.3 では無視 */
  UINT8 ch; /* TWDM における下り波長チャネル ID. 非対応の場合は無視 */
  int n_of_configs;  /* 本 API で通知する送信許可の個数 */
  grant_config_t grant_config[];  /* 送信許可（n_of_configs の個数の配列） */

  typedef struct{              /* IEEE802.3          ITU-T G.989.3    */
   UINT16 id;                  /* LLID               Alloc-ID */
   UINT8  flags;               /* Flags              Flags/FWI/Burst Profile */
   UINT32 grant_start_time;    /* Grant Start Time   Start Time */
   UINT16 grant_length;        /* Grant Length       GrantSize */
  }grant_config_t;
```

説明：

本 API により，API 上側にある DBA 機能部は，API 下側の DBA 処理部に送信許可量を直接的に通知する．DBA 処理部は，通知された送信許可量の情報をもとに，ONU への送信許可メッセージを組み立て，ONU に送信する．

以下，IEEE802.3 の場合と，ITU-T G.989 の場合とで，それぞれ詳しく動作を記す．

＜IEEE802.3＞

IEEE802.3 Ethernet PON では，上り送信制御は GATE メッセージを ONU に送ることで行う．宛先となる ONU はプリアンブルに格納された LLID により識別される．送信開始時刻は grant start time，送信許可量は grant length によって指示される．送信許可の種類として，Discovery GATE，force report のフラグフィールドがある．また，１つの GATE メッセージにより最大４つまでの送信許可を格納することができる．

本 API を受けた処理側としては，引数を parse し，以下のように動作することを期待す

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

る.

・ sfc, ch の値は無視する.
・ １つの grant_config が１つの送信許可（grant start time と grant length の組）に相当し, n_of_configs 個数だけある.
・ id の下位 15bit を GATE に付与する LLID とする.
・ flags の最下位 bit は discovery flag, 2bit 目は force_report の値とする.
・ grant__start_time は, 32bit すべてをそのまま Grant Start Time の値とする.
・ grant_length は, Grant Length の値とする.
・ １つの grant_config について, 上記のように１つの grant とする. １つの id (LLID) に対して複数の grant_config がある場合には, なるべく１つの GATE メッセージに詰め込む. GATE メッセージには最大４つまでの grant を詰め込むことができる. また, GATE メッセージにおける number of grants の値は, API 下側の処理部が詰めた GATE メッセージを元に算出し, 値を格納する. また, force_report の値は, 当該 grant が何番目の grant であるかをもとに, API 下側の処理部で算出し, 値を格納する.
・ それ以外の GATE フレームの各フィールドの値については, API 上側からは指定しない.
・ API を受け, 引数の parse が完了したのち, 完全に構築された GATE フレームから直ちに下り送信する.

なお, API 上部にとって, 現在の MPCP ローカルタイム値, ONU の識別, LLID 番号・RTT 値の取得, リンク状態の取得などは他のプロセスにより実施されていることを前提としている.

同様に, ONU/LLID ごとの QoS パラメータ（最大帯域等）も, API 上部の DBA アプリには他のプロセスにより通知されていることを前提としている.

＜G.989.3＞

ITU-T G.989.3 においては, 上り送信制御は BWmap を各 ONU に通知することで行われる. BWmap は複数の allocation structure から構成され, １つの allocation structure に１つの送信許可が含まれる. 送信許可は IEEE802.3 とよく似ており, StartTime と GrantSize から構成される.

本 API を受けた処理側としては, 引数を parse し, 以下のように動作することを期待する.

・ 受けた sfc の値と等しい super frame counter の下りフレームの BWmap に搭載する.
・ ch の値が示す DWLCH ID の下り波長チャネルで BWmap を下り送信する. TWDM に非対応の場合は, この値は無視する.

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

・ １つの grant_config が１つの Allocation structure に相当し，n_of_configs が Allocation structure の個数を表す．
・ id の下位 14bit を Allocation structure に付与する Alloc-ID とする．
・ flags の最下位 bit は，Allocation structure 中の Flags における，PLOAMu，2bit 目は DBRu とする．
・ また，同様に flags の 3bit 目は FWI，4-5bit 目は Burst Profile の値とする．
・ grant__start_time の下位 32bit を StartTime の値とする．
・ grant_length は，GrantSize の値とする．
・ １つの grant_config について，上記のように１つの Allocation structure とする．Allocation Structure 中の HEC については，API 下側の処理部で計算し格納する．
・ １つの API につき，１つの BWmap を構築するものとする．
・ API を受け，BWmap を構築したのち，本 API により指定された superframe counter 値の下りフレームに合わせて FS ヘッダに含めて BWmap を下り送信する．

なお，API 上部にとって，現在の superframe counter 値，ONU の識別，Alloc-ID 番号の紐づけ，RTT 値の取得，リンク状態の取得などは他のプロセスにより実施されていることを前提としている．

同様に，Alloc-ID ごとの QoS パラメータ（最大帯域等）も，API 上部の DBA アプリには他のプロセスにより通知されていることを前提としている．

5.2.1.5.2　送信要求量取得 API

形式：

fasa_api_get_onu_request(UINT64 sfc, UINT8 ch, int n_of_configs, request_config_t request_config[]);

引数：

```
UINT64 sfc;  /* superframe カウンタ値 IEEE802.3 では無視 */
UINT8 ch; /* TWDM における上り波長チャネル ID. 非対応の場合は無視 */
int n_of_requests;  /* 本 API で通知される送信要求の個数 */
request_config_t request_config[];  /* 送信要求（n_of_configs の個数の配列） */

typedef struct{              /* IEEE802.3          ITU-T G.989.3  */
 UINT16 id;                  /* LLID               ONU-ID */
```


Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

```
  UINT8  flags;              /* QSet/Qreport 番号    Ind */
  UINT32 request;            /* queue report 値      BufOcc 値*/
 }request_config_t;
```

説明：

本 API により，API 上側にある DBA 機能部は，API 下側の DBA 処理部にて受信，蓄積されていた送信要求に関する情報を直接的に取得する．本 API は，ここではコールバックではなく，ポーリングの形式としている．

以下，IEEE802.3 の場合と，ITU-T G.989.3 の場合とで，それぞれ詳しく動作を記す．

＜IEEE802.3＞

IEEE802.3 Ethernet PON では，上り送信要求は REPORT メッセージを各 ONU が OLT に送ることで行う．送信元となる ONU はプリアンブルに格納された LLID により識別される．

REPORT フレームには，Queue Set と呼ぶ，Report bitmap と Queue Report の組が１つ以上含まれる．Queue Set の個数は number of queue sets で表される．

送信要求量は Queue Report に値が格納される．１つの Queue Set には，最大８種類の Queue Report を格納することができ，値のある Queue Report のみを通知できる．具体的には，Report bitmap により，８種類のうち，どの Queue Report を通知したのか示している．

本 API を受けた処理側としては，引数の戻り値として，送信要求に関する情報を返すとともに，返すべく，以下のように動作することを期待する．

- 受信した REPORT フレームに含まれる送信要求情報を蓄積する．具体的には，LLID，Queue Set 番号，Queue Report 番号と，これらの番号が示す queue report 値を蓄積する．
- この３つを API の引数 request_config の戻り値として，API 上部に返す．
- 具体的には，LLID の値は引数 id に格納する．
- Queue report 番号 0-7 は，引数 flags の下位 3bit に格納する．
- Queue Set 番号は引数 flags の上位 5bit に格納する．
- これらの番号に対応した queue report の値を引数 request に格納する．
- 蓄積していた送信要求の情報は，本 API による読み出しに応じて API 上側に引き渡し，引き渡した情報は，消去するか，新たな情報で上書きしていく．
- 引数 sfc には，蓄積している送信要求情報のうち，最も時間的近傍に REPORT フレームを受信した際の MPCP ローカルタイムを格納する．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

なお，API 上部にとって，現在の MPCP ローカルタイム値，ONU の識別，LLID 番号・RTT 値の取得，リンク状態の取得などは他のプロセスにより実施されていることを前提としている．

同様に，ONU/LLID ごとの QoS パラメータ（最大帯域等）も，API 上部の DBA アプリには他のプロセスにより通知されていることを前提としている．

＜ITU-T G.989.3＞

ITU-T G.989.3 では，上り送信要求は DBRu 中の BufOcc を各 ONU が OLT に送ることで行う．送信元となる ONU は FS ヘッダに格納された ONU-ID により識別される．また，FS ヘッダ中の Ind フィールドにおける PLOAM queue status ビットにより，上り PLOAM メッセージの送信待ちの有無を ONU は OLT に通知する．

本 API を受けた処理側としては，引数の戻り値として，送信要求に関する情報を返すとともに，返すべく，以下のように動作することを期待する．

- 受信した送信要求情報を蓄積する．具体的には，ONU-ID，BufOcc 値，PLOAM queue status ビット値を蓄積する．
- この３つを API の引数 request_config の戻り値として，API 上部に返す．
- 具体的には，ONU-ID の値は引数 id に格納する．
- PLOAM queue status ビット値は，引数 flags の最下位ビットに格納する．
- BufOcc 値は引数 request に格納する．
- １つのバースト中に複数の Allocation があった場合には，受信した順に BufOcc 値を蓄積する．このとき，ONU-ID 値，PLOAM queue status はそれぞれの BufOcc 値に対して同じ値となり，情報としては冗長だが，API 引数のシンプルさ，統一を優先する．
- 蓄積していた送信要求の情報は，本 API による読み出しに応じて API 上側に引き渡し，引き渡した情報は，消去するか，新たな情報で上書きしていく．
- 引数 sfc には，蓄積している送信要求情報のうち，最も時間的近傍に BufOcc を受信した際の Superframe counter 値を格納する．

なお，API 上部にとって，現在の superframe counter 値，ONU の識別，Alloc-ID 番号の紐づけ，RTT 値の取得，リンク状態の取得などは他のプロセスにより実施されていることを前提としている．

同様に，Alloc-ID ごとの QoS パラメータ（最大帯域等）も，API 上部の DBA アプリには他のプロセスにより通知されていることを前提としている．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

5.3. L2 主信号処理機能

OLT における L2 主信号処理としては，上り下りそれぞれの方路へ適切にユーザデータを転送することにある．そのため，API 上部側の役割としては，EMS/上位 OpS からの Netconf/YANG，あるいは Openflow による指示を受信し，この指示に基づき，(1)上り下り方路それぞれへの転送設定，(2)統計情報の取得，(3)ONU への転送設定を API 下部側へ展開する．(1),(2)は YANG モデルに基づき API 下部側へ設定を展開する処理となり，(3)は ONU への設定内容を組み立て,ONU へのメッセージ送信指示を API 下部側へ展開する処理となる．

~~TBD~~

5.4. 保守運用機能

OLT における保守運用機能としては，多くの機能がありうるが，以下の２つに大別できる．(1) OLT への設定・動作指示，(2)OLT および ONU の状態通知．
(1)設定・動作指示においては，API 上部側は，EMS/上位 OpS からの Netconf による指示を受信し，YANG モデルに基づいて内容を API 下部側へ展開する．一方で，(2)状態通知においては，API 上部側は，YANG モデル，あるいは OAM/OMCI メッセージに基づく API 下部側からの通知を受け取り，Netconf により EMS/上位 OpS へその内容を通知する．

~~TBD~~

5.5. PON マルチキャスト機能

主に映像配信などに用いられる PON マルチキャスト機能には，いくつかの実現方式がある．本節では，それらの方式の概要を説明するとともに，API 上部，下部での機能分担，メッセージフローのイメージを示す．

5.5.1 PON マルチキャスト方式概説

マルチキャストは，任意の複数の転送先（１つの場合もある）に，同じ情報を同報する方式である．一般に，端末からのマルチキャストグループへの参加要求と，離脱要求に応じて，マルチキャストストリームの転送先が動的に制御される．参加要求・離脱要求などのメッセージ，マルチキャスト転送制御のためのプロトコルは，IPv4 の IGMPv3，IPv6 の MLDv2 が用いられる場合が多い．ここで，TDM 系の PON では，OLT から ONU への下り方路は，一般に論理的にはユニキャスト，物理的にはブロードキャストとなっているため，マルチキャストを実現するには工夫が必要である．

その方法として，主には，以下の 3 つがある．(1)上位ノードによるマルチキャスト，(2)ONU スヌープ，(3)OLT プロキシ．これら３つの方式について，それぞれ機能分担，メッセージフローのイメージを示す．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

5.5.2 上位ノードによるマルチキャスト

上位ノードにより，マルチキャスト転送を実現する方式の場合，ONU および OLT は IGMP/MLD メッセージをそれぞれ透過転送するように設定する．そうして参加要求メッセージを受信した OLT より上位のノードにより，参加要求を発した端末に宛ててマルチキャストストリームが転送される．このとき，同一 OLT につながる複数の ONU 配下の端末から，ある同じマルチキャストグループにそれぞれ参加要求があったとすると，上位ノードはそれぞれの端末に向けてマルチキャストストリームを転送するため，同一内容の複数のストリームが OLT に送信されることになる．OLT は，それら複数のストリームを，個別のユニキャストストリームとして，それぞれの ONU に透過転送する．

また，同一 ONU 配下の複数の端末から，同じマルチキャストグループのへの参加要求があった場合は，ONU および配下ノードの機能構成によって異なる．ONU あるいは配下ノードがマルチキャストルータ機能を持っている場合には，２台目の端末からの参加要求に対して，OLT および上位へ参加要求メッセージを転送することなく，ONU あるいは配下ノードがマルチキャストストリームを２台目の端末へ同報配信する．

マルチキャストルータ機能を備えない構成の場合には，それぞれの端末に対するマルチキャストストリームが OLT より上位のノードにより配信されることになる．

5.5.3 ONU スヌープ

ONU を流れる IGMP/MLD メッセージを覗き見ることで，PON マルチキャストを実現する方法もある．一般に，ONU スヌープと呼ばれる．この方法では，ONU 配下の端末から，OLT より上位のノード（マルチキャストルータ）に送信される IGMP/MLD メッセージを，ONU にて覗き見ることにより，PON マルチキャストを実現するため，このように呼ばれる．

この方法では，まず上位ノードから受信するマルチキャストストリームは全 ONU が受信できるように OLT は転送しておく．

ONU では，覗き見た（スヌープした）IGMP/MLD メッセージに応じて，自らの下り転送フィルタを開閉する．詳細には，スヌープしたメッセージが参加要求であれば，参加するマルチキャストグループのトラヒックを下り転送するよう転送フィルタを設定する．一方で，スヌープしたメッセージが離脱要求であれば，遮断するように転送フィルタを設定する．転送・遮断のフィルタ設定は，IP アドレス，MAC アドレス，VLAN タグ，他の識別子など，いろいろな方法があり，予め定められた方法で行う．

これにより，ONU のフィルタが開いた状態であれば，OLT から転送されてきたマルチキャストストリームを ONU 下部へ転送することができ，フィルタが閉じた状態であれば，ONU が OLT から受信したマルチキャストストリームは ONU 下部へ転送されることなく，廃棄される．これにより，マルチキャスト転送を実現している．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

このとき，API 上部，下部での機能分担としては，EMS/上位 OpS からの Netconf 等による初期設定，あるいはサービスオーダにより ONU の IGMP/MLD スヌープ機能の有効・無効の指示を API 上部が受ける．これを受けて，API 下部側に対しては，ONU とのコミュニケーション API を介して，拡張 OAM あるいは OMCI メッセージの送信を指示する．API 下部側は，指示を受けたメッセージを ONU に対して送信し，スヌープ機能の有効・無効を指示する．これにより，ONU スヌープによる PON マルチキャストが制御される．

5.5.4 OLT プロキシ

ONU から OLT を経由して上位ノードに転送される IGMP/MLD メッセージを，OLT が集約し代理応答しつつ，ONU に対して下り転送フィルタの開閉を指示する方法もある．この方法は，一般に OLT プロキシと呼ばれる．

この方法でも，上位ノードからのマルチキャストストリームは，全 ONU が受信できるように OLT は転送しておく．

ONU 下部の端末からの IGMP/MLD メッセージは，OLT が一度受信し，メッセージ内容に応じて上位ノードに転送する．また，メッセージが参加要求であれば，OLT は拡張 OAM もしくは OMCI メッセージにより，ONU の該当するマルチキャストグループの下り転送フィルタを開くよう ONU に指示する．一方，離脱要求であれば，同様に，下り転送フィルタを閉じるよう ONU に指示する．これにより，参加要求のあった端末宛てにのみマルチキャストストリームが転送されることで，マルチキャスト転送を実現している．

この際，ONU 配下に複数の端末がある場合や，IGMP/MLD メッセージを転送してきた ONU とは異なる ONU 配下の端末の状態などを考慮して，効率的にマルチキャスト転送されるような ONU のフィルタ操作，上位ノードへのメッセージ転送を行うこともできる．

このとき，API 上部，下部での機能分担としては，まず，API 上部側は，ONU から上り送信された IGMP/MLD メッセージを，OLT が受信したのち API 上部側へ転送するよう，予め主信号の方路設定を行う．この方路設定は，API 上部から下部への OLT への主信号設定の一環であり，Netconf/YANG として，あるいは Openflow として設定されることを想定している．方路設定自体のトリガは，EMS/上位 OpS からの設定になる．

また，マルチキャストストリームの下り転送の方法も，EMS/上位 OpS からの設定指示を NC/Y もしくは Openflow により API 上部が受け，その内容を API 下部に展開することで行う．OLT プロキシ機能としては，API 上部へ転送された IGMP/MLD メッセージの内容に基づき，API 下部に対して ONU の下りフィルタの開閉を指示する拡張 OAM もしくは OMCI メッセージの送信を指示することで実現される．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

~~TBD~~

5.6. 省電力制御機能

省電力制御機能としては，必要に応じて ONU が一部機能への給電を停止し，ONU での消費電力を低減するものである．API 上部側の役割としては，EMS/上位 OpS から ONU の省電力モードに関する設定，サービスオーダを受信し，内容に基づく拡張 OAM/OMCI メッセージを組み立て，API 下部側にこのメッセージを ONU に送信するよう通知する．

また，PLOAM などによる状態の変化通知を，API 下部側から API 上部側は受信する．

一方で，上記の DBA と同様に，API 上部側から直接的に ONU の省電力モードの状態を制御したい場合もありうる．このとき，ONU への送信メッセージの組み立て，ONU からの受信メッセージの取り込みを，リアルタイム性を持って API 上部側で行い，それぞれ，API 下部側へメッセージ送信，メッセージ受信指示を行う．

~~TBD~~

5.7. 周波数/時刻同期機能

周波数/時刻同期機能においては，OLT に入力された基準信号や時刻情報を，PON 区間を介して ONU から正確に出力させる機能である．

API 上部側の役割としては，同期機能に必要となる設定や，OLT から ONU への信号伝搬に可関わるパラメータ等を ONU に通知するために，送信メッセージを組み立て，API 下部側へメッセージ送信の指示を行う．

~~TBD~~

5.8. プロテクション機能

TBD

5.9. 外部連携機能

上に記載したモバイル基地局との低遅延化 DBA のように，外部の機器との連携により機能実行する場合もある．このような外部連携機能を実現するためには，特に API 上部側による外部の機器からのメッセージを受信出来ることが必要である．

一方で，この外部機器からのメッセージ受信機能は，実装・外部機器との接続構成・メッセージ形式に強く依存するため，基本的には，API 上部側の役割としては，メッセージを分解せずに受信し，parse することを想定している．また，搭載 OS の標準的な機能等を活用できると考えられるが，今後の検討の進展により，必要があれば独自の API を規定する．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.

6. まとめ

本資料では，我々の提案するFASAコンセプトについて示した．また，それを実現するためのアーキテクチャ・論理モデルおよび，分担した機能間のインターフェースを示した．

Copyright(c) 2016-2017 NTT Corp. All Rights Reserved.