Data-Over-Cable Service Interface Specifications
Flexible MAC Architecture

FMA MAC Manager Interface Specification

CM-SP-FMA-MMI-I03-220126

ISSUED

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	CM-SP-FMA-MMI-I03-220126			
Document Title:	FMA MAC Manager Interface Specification			
Revision History:	W01 – Released 03/22/2019 W02 – Released 07/24/2019 W03 – Released 10/23/2019 D01 – Released 01/30/2020 D02 – Released 05/04/2020 D03 – Released 08/18/2020 I01 – Released 09/30/2020 I02 – Released 05/26/2021		I03 – Released 01/26/2022	
Date:	January 26, 2022			
Status:	~~Work in Progress~~	~~Draft~~	**Issued**	~~Closed~~
Distribution Restrictions:	~~Focus Group Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A generally public document that has undergone Member and Technology Supplier review, cross-vendor interoperability, and is for Certification testing if applicable. Issued Specifications are subject to the Engineering Change Process.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

Contents

Figures

Tables

1 SCOPE

1.1 Introduction and Purpose

The MAC Manager Interface (MMI) is the cornerstone of the FMA initiative as it defines the method of communication to enable interoperability between the MAC Manager and MAC Network Elements from different suppliers. The MMI will sometimes be referred to as the southbound interface of the MAC Manager. The MMI sets itself apart from hardware abstraction interfaces in that there will not be different interfaces defined for each vendor's MAC Network element, such as RMD, RMC, pCore, and vCore. This interface operates at a higher layer so that the same interface may be used regardless of MAC Manager and MAC-NE supplier. The MMI is based on RESTCONF/YANG. The MAC Manager Interface is one of the core objectives of the FMA initiative. This specification describes the functional requirements of the MAC Manager and MMI protocol to manage MAC Network element. The MAC Manager northbound interface, called the FMA-OSSI, is documented in another specification called the [FMA-OSSI].

1.2 FMA Interface Documents

A list of the documents in the FMA family of specifications is provided in Table 1. For the current versions of these specifications, refer to http://www.cablelabs.com/specs/specification-search/.

Table 1 - List of FMA Specifications

Designation	Title
CM-SP-FMA-SYS	Flexible MAC Architecture System Specification
CM-SP-FMA-MMI (this document)	Flexible MAC Architecture MAC Manager Interface Specification
CM-SP-FMA-PAI	Flexible MAC Architecture PacketCable™ Aggregator Interface Specification
CM-SP-FMA-OSSI	Flexible MAC Architecture Operations Support System Interface Specification

1.3 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This word means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood, and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood, and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

1.4 Conventions

In this specification the following convention applies any time a bit field is displayed in a figure. The bit field can be interpreted by reading the figure from left to right, then from top to bottom, with the Most Significant Bit (MSB) being the first bit to read and the Least Significant Bit (LSB) being the last bit so read.

`XML Schema and YANG module syntax are represented by this code sample font.`

NOTE: Notices and/or Warnings are identified by this style font and label.

1.5 Organization of Document

Section 1 provides an overview of the MAC Manager MMI specification including a list of the FMA family of specifications.

Section 2 includes a list of normative and informative references used within this specification.

Section 3 defines the terms used throughout this specification.

Section 4 defines the acronyms used throughout this specification.

Section 5 provides an introduction to the FMA architecture specific to the MAC Manager and MAC-NE MMI interface.

Section 6 defines the MAC Manager to MAC-NE initialization process for MMI.

Section 7 defines the MMI method of operations.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

All references are subject to revision, and parties to agreement based on this specification are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

[FMA-SYS]	DOCSIS Flexible MAC Architecture System Specification, CM-SP-FMA-SYS-I03-220126, January 26, 2022, Cable Television Laboratories, Inc.
[FMA-YANG]	Flexible MAC Architecture, YANG Modules. http://mibs.cablelabs.com/YANG/DOCSIS/FMA
[RFC 4122]	IETF RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace, July 2005
[RFC 4648]	IETF RFC 4648, The Base16, Base32, and Base64 Data Encodings, October 2006
[RFC 5246]	IETF RFC 5246, Transport Layer Security (TLS) Protocol Version 1.2, August 2008
[RFC 5424]	IETF RFC 5424, The Syslog Protocol, March 2009
[RFC 6020]	IETF RFC 6020, YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), October 2010
[RFC 6241]	IETF RFC 6241, Network Configuration Protocol (NETCONF), June 2011
[RFC 6585]	IETF RFC 6585, Additional HTTP Status Codes, April 2012
[RFC 7231]	IETF RFC 7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, June 2014
[RFC 7232]	IETF RFC 7232, Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests, June 2014
[RFC 7895]	IETF RFC 7895, YANG Module Library, June 2016
[RFC 7950]	IETF RFC 7950, The YANG 1.1 Data Modeling Language, August 2016
[RFC 8040]	IETF RFC 8040, RESTCONF Protocol, January 2017
[RFC 8407]	IETF BCP 216/RFC 8407, Guidelines for Authors and Reviewers of Documents Containing YANG Data Models, October 2018
[RFC 8446]	IETF RFC 8446, The Transport Layer Security (TLS) Protocol Version 1.3

2.2 Informative References

This specification uses the following informative references.

[FMA-OSSI]	Flexible MAC Architecture Operations Support System Interface, CM-SP-FMA-OSSI-I01-211101, November 1, 2021, Cable Television Laboratories, Inc.
[FMA-PAI]	DOCSIS FMA PacketCable Aggregator Interface Specification, CM-SP-FMA-PAI-I01-200930, September 30, 2020, Cable Television Laboratories, Inc.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; http://www.cablelabs.com
- Internet Engineering Task Force (IETF) Secretariat, 48377 Fremont Blvd., Suite 117, Fremont, California 94538, USA; Phone: +1-510-492-4080; Fax: +1-510-492-4001; http://www.ietf.org

3 TERMS AND DEFINITIONS

3.1 Terms and Definitions

This specification uses the following terms.

Cable Modem	A modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system
Dynamic Host Configuration Protocol (DHCP)	A network protocol enabling a server to automatically assign an IP address to a network element
Flexible MAC Architecture	The distributed CMTS/CCAP architecture where the DOCSIS MAC and PHY layer processing of a CMTS are disaggregated and moved deeper into the access network
MAC Manager	1) management entity that represents a single point of contact for operator back-office management systems to FMA access network elements within its scope of control and 2) FMA functional entity that uses FMA protocols to manage an RMD
Media Access Control (MAC)	Refer to the Layer 2 element of the system which would include DOCSIS framing and signaling.
Network Configuration Protocol	An IETF network management protocol that provides mechanisms to manipulate the configuration of a device, commonly referred to as NETCONF. NETCONF executes YANG-based XML files containing configuration objects.
PacketCable Aggregator (PAG)	Control plane entity that represents a single point of contact for operator back-office PacketCable and Policy Server systems to FMA access network elements within its scope of control
PacketCable Aggregator Interface (PAI)	Control plane interface between a PAG and subtended MAC-NEs
Remote MAC Core (RMC)	A Device in the network that implements the FMA specifications to provide MHAv2 DOCSIS Core functionality closer to the RPD than is possible in the MHAv2 architectural model
Remote-MACPHY Device (RMD)	A Device in the network that implements the FMA specifications to provide conversion from digital Ethernet transport to analog RF transport
Remote-PHY Device (RPD)	A Device in the network that implements the Remote-PHY specification to provide conversion from digital Ethernet transport to analog RF transport
RESTCONF	An HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG, using the datastore concepts defined in the Network Configuration Protocol (NETCONF)
RESTCONF Client	A client application, residing in the MAC Manager, that implements the RESTCONF protocol
RESTCONF Server	A server application, residing in a MAC-NE, that implements the RESTCONF protocol
Trivial File Transfer Protocol (TFTP)	A file transfer protocol generally used for automated transfer of configuration or boot files between machines
YANG	A data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations.

CLI	Command Line Interface
CM	Cable Modem
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Servers
DOM	Digital Optical Monitoring
ETAG	Entity-Tag
FCAPS	Fault, Configuration, Accounting, Performance, Security
FMA	Flexible MAC Architecture
HTTP	HyperText Transfer Protocol
IPDR	Internet Protocol Detail Record
JSON	JavaScript Object Notation
MAC	Media Access Control
MAC-NE	MAC Network Element
MM	Multimedia
MMI	MAC Manager Interface
NETCONF	Network Configuration Protocol
OCC	Optimistic Concurrency Control
OUI	Organizationally Unique Identifier
PAI	PacketCable™ Aggregator Interface
PMA	Profile Management Application
PNM	Proactive Network Maintenance
RESTCONF	REpresentational State Transfer Configuration Protocol
RMC	Remote MAC Core
RMD	Remote MACPHY Device
RP	Roll-off postfix
RPC	Remote Procedure Call
RPD	Remote PHY Device
SNMP	Simple Network Management Protocol
SP	Streaming Protocol
SRV	Service
SSH	Secure Shell
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
ToD	Time of Day
UML	Unified Modeling Language
URI	Uniform Resource Identifier

5 ARCHITECTURE OVERVIEW

Figure 1 represents the FMA reference architecture as defined in [FMA-SYS]. This specification defines the MAC Manager Interface (MMI), labeled as the "Mm-MacNe" interface in Figure 1.

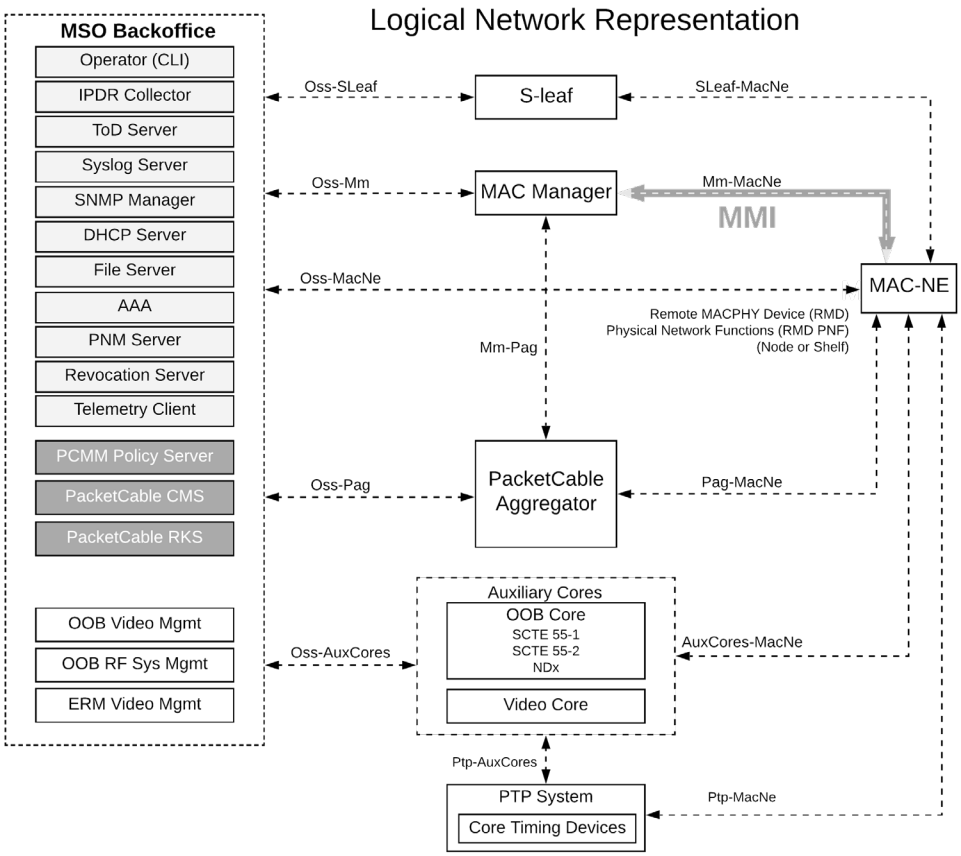

Figure 1 - FMA Phase 1 Reference Architecture

5.1 MAC Manager Architecture

The primary role of the MAC Manager is to serve as an interface point for management plane functions of the MAC-NE as illustrated in Figure 2. The MAC Manager is an aggregation point for what will be a large volume of MAC-NEs in order to make the many appear as one to the back-office systems.

A MAC Manager could have a 1:1 relationship to a MAC-NE or a MAC Manager may have a 1:N relationship with many MAC-NEs. A MAC-NE could have a 1:1 relationship to a MAC Manager or a 1:N relationship with multiple MAC Managers.

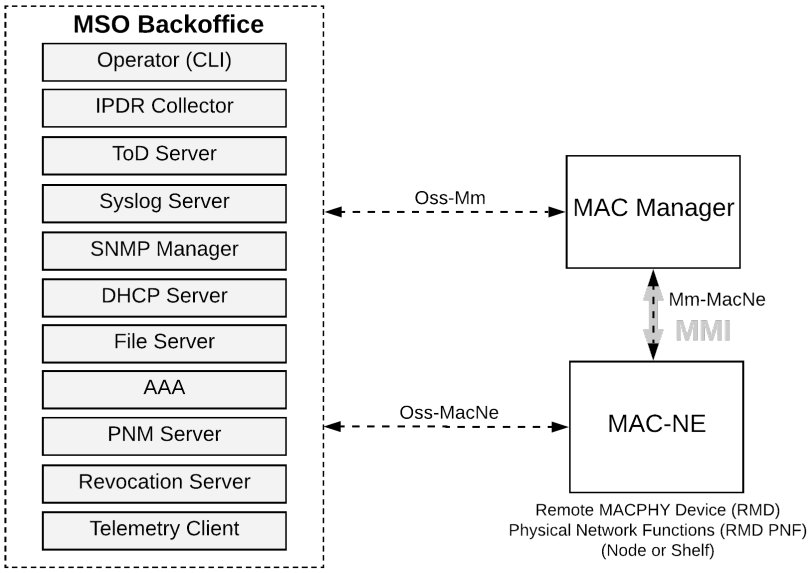

Figure 2 - FMA Management Overview

The MAC Manager Northbound Interface ("Oss-Mm"), referred to as FMA OSSI, implements all defined OSSI management interfaces including but not limited to SNMP, SNMP Notifications/Traps, Syslog, and IPDR while NETCONF is optionally specified for configuration management. In addition, the MAC-NE Northbound OSS Interface ("Oss-MacNe") implements OSSI interfaces including Syslog and Streaming Telemetry. Refer to [FMA-OSSI] for additional details.

The MAC Manager Southbound Interface ("Mm-MacNe"), referred to as MAC Management Interface (MMI), enables the MAC Manager to support [FMA-OSSI] interfaces such as: SNMP, SNMP Notifications/Traps, Syslog, and support for legacy and vendor CLI commands. Refer to [FMA-OSSI] for a detailed MAC Manager Management Architecture. The YANG models are not defined within this document and are available directly from CableLabs at [FMA-YANG]. The MAC-NE hosts the server-side of MMI while the MAC Manager acts as an MMI client.

The interfaces between the MAC Manager and the FMA MAC-NE device are described in this specification.

6 MAC MANAGER TO MAC-NE INITIALIZATION PROCESS

When MAC Network Elements are attached to the network and power on, they discover their MAC Manager and then establish communications using the following Southbound Interface. This process is documented in [FMA-SYS].

7 MMI METHODS OF OPERATIONS

The MAC Manager serves as an aggregation point for one or more FMA MAC-NE devices for management purposes. The MAC Manager northbound exposes all defined [FMA-OSSI] management interfaces. The MAC Manager southbound, MMI, defines a data model and interface that allows a MAC Manager to communicate with a MAC-NE to gather the data required to fulfill [FMA-OSSI] and other requirements.

The MAC Manager MUST act as an MMI client (RESTCONF Client), issuing MMI read and write operations to one or more MMI servers.

The MAC-NE MUST act as an MMI server (RESTCONF Server), fulfilling MMI read and write operations for one or more MMI clients.

Interfaces between the MAC Manager and the MAC-NE are defined using YANG-based data models. To manage that data model, RESTCONF has been chosen as the protocol for creating, reading, updating, and deleting instances of YANG objects between the MAC Manager and the FMA device. The YANG models are not defined within this document and are available directly from CableLabs at [FMA-YANG].

The MMI YANG data models support the full FCAPS model and is serviced by the MAC-NE RESTCONF protocol and corresponding API. The MMI data models provide a unified, object-oriented configuration and status view of the MAC-NE.

7.1 MMI Operations (Informative)

7.1.1 Read Operations

RESTCONF supports reading configuration and status objects through the defined MMI YANG data model objects. YANG defines the structure of the data model, while RESTCONF provides a framework for encoding and transmitting the objects.

The MAC Manager reads YANG data models using the RESTCONF HTTP verb 'GET'. The HTTP URI provides a hierarchical selector that allows the MMI client to request a subset of the full YANG data model.

7.1.2 Write Operations

RESTCONF supports writing configuration objects through the defined MMI YANG data model objects. YANG defines the structure and constraints of the configurable data models, while RESTCONF provides a framework for encoding and transmitting the objects.

The RESTCONF HTTP verb provides a top-level success or error code and the HTTP payload contains details of the error codes.

The MAC Manager writes configuration to the MAC-NE using the RESTCONF HTTP verb 'PUT' or 'POST'.

The MAC Manager deletes configuration from the MAC-NE using the RESTCONF HTTP verb 'DELETE'.

7.1.3 RPC Operations

RESTCONF/YANG support defining Remote Procedure Call (RPC) interfaces. RPCs are action-oriented, not object-oriented. They provide a mechanism for the MMI client (on the MAC Manager) to request the MMI server (on the MAC-NE) to take an action and report the result of that action. This contrasts with the read/write operations which read and write hierarchical data models.

7.1.4 Notification Operations

RESTCONF/YANG support defining asynchronous notification streams (similar to SNMP Traps). Notifications are to be subscribed to by the MMI client (on the MAC Manager). The MMI server (on the MAC-NE) will asynchronously push notifications to any subscribed MMI client. YANG data models define the notifications that can be subscribed to, as well as the payload to expect on that notification stream.

7.2 RESTCONF

The MAC Manager MUST implement a Southbound Interface to a MAC-NE over the RESTCONF [RFC 8040] protocol to support a model driven method of management and device provisioning. RESTCONF is a REST based approach to NETCONF datastore control. RESTCONF provides for all required MAC Manager operations with MAC-NEs.

A model driven method simply stated is an API that may be generated from a YANG model describing the MAC-NE capabilities. RESTCONF defines how a YANG model effectively maps directly to an object class hierarchy over RESTful interface for use by the MAC Manager.

The MAC Manager management and control of MAC-NE supported data model is described by YANG and uses RESTCONF as a protocol carrying JSON encoded data over HTTPS transport to the MAC-NE. The RESTCONF standard builds upon datastore concepts defined from NETCONF standards.

RESTCONF uses HTTP methods for all CREATE, READ, UPDATE, DELETE (CRUD) operations:

Table 2 - HTTP Methods

Option	Method
GET	Read
PATCH	Update
PUT	Create / Update
POST	Create
DELETE	Delete
HEAD	Header Metadata

The MAC-NE MUST implement a RESTCONF [RFC 8040] Server.

The MAC-NE MUST implement an HTTP [RFC 7231] server to provide RESTCONF [RFC 8040] functionality.

7.2.1 Request Header

The MAC-NE expects standard HTTP Header behavior.

The MAC Manager MUST include an 'Accept:' header with the value of 'application/yang-data+json' when issuing a GET.

The MAC Manager MUST include a 'Content-Type:' header with a value of 'application/yang-data+json' when issuing a PUT, POST, or PATCH.

The MAC-NE MUST provide JSON response payloads when the HTTP 'Accept:' header has the value 'application/yang-data+json'.

The MAC-NE MAY provide XML response payloads if the HTTP 'Accept:' header has the value 'application/yang-data+xml'.

The MAC-NE MUST reject any 'Accept:' header that it cannot process with an HTTP status code of 406 Not Acceptable as defined in [RFC 7231].

The MAC-NE MUST reject any 'Content-Type:' header that it cannot process with an HTTP status code if 406 Not Acceptable as defined in [RFC 7231].

7.2.2 Response Header

The MAC Manager MUST support receiving any arbitrary well-formed HTTP response header.

The MAC Manager MAY ignore response headers it does not understand.

The MAC-NE MAY provide any standard or non-standard well-formed HTTP response header(s).

The MAC-NE MUST populate the 'Content-Type:' header with the value of 'application/yang-data+json'.

The MAC-NE MUST populate the 'Date:' header with the MAC-NE's local clock.

7.2.3 ETAG and Last-Modified Header

The MAC-NE MUST support Entity-Tag (ETAG) requirements detailed in [RFC 8040]; this is a strengthening of the base RFC requirements.

The MAC-NE SHOULD support Timestamp / Last-Modified requirements detailed in [RFC 8040]; this is a strengthening of the base RFC requirements. If a MAC-NE supports Timestamp / Last-Modified, the MAC-NE MUST accept and support the HTTP headers 'If-Modified-Since' and 'If-Unmodified-Since', described in [RFC 7232].

The MAC-NE MUST reject all RESTCONF update or change write operations (PUT, POST, PATCH) on a resource that already exists on the MAC-NE which does not contain an 'If-Match' HTTP header with an HTTP status code of 428 Precondition Required from [RFC 6585].

The MAC-NE MUST reject all RESTCONF update or change write operations (PUT, POST, PATCH) on a resource where the MAC Manager included 'If-Match' HTTP header value does not match the current ETAG of the resource on the MAC-NE with an HTTP status code of 412 Precondition Failed from [RFC 7231].

7.2.4 CableLabs Request and Response Headers

The MAC-NE MUST populate the following HTTP Headers in every HTTP query and response. This includes the "Announce" functionality detailed in [FMA-SYS].

- The MAC-NE MUST populate an 'X-CL-DEVICE:' HTTP Header with "RMD" for Remote-MACPHY Devices and "RMC" for Remote-MAC-Core Devices.
- The MAC-NE MUST populate an 'X-CL-VENDOR:' HTTP Header with the OUI of the MAC-NE's Vendor, where the OUI bytes are to be separated by colons.
- The MAC-NE MAY populate an 'X-CL-VENDOR-STRING:' HTTP Header with a vendor-defined string representing the vendor name.
- The MAC-NE MUST populate an 'X-CL-SW-VERSION:' HTTP Header with the Software Version of the MAC-NE. The exact format of the SW Version string is vendor-specific.
- The MAC-NE MUST populate an 'X-CL-HW-VERSION:' HTTP Header with the Hardware Version of the MAC-NE. The exact format of the HW Version string is vendor-specific.
- The MAC-NE MUST populate an 'X-CL-HW-IDENTIFIER:' HTTP Header with the MAC-NE Unique Identifier as defined in [FMA-YANG].
- The MAC-NE MUST populate an 'X-CL-SERIAL-NUMBER:' HTTP Header with the MAC-NE's serial number. The exact format of the serial number is vendor-specific.

The MAC Manager also needs to populate certain HTTP Headers, as follows:

- The MAC Manager MUST populate an 'X-CL-DEVICE:' HTTP Header with "MAC Manager" in every HTTP query and response.
- The MAC Manager MUST populate an 'X-CL-VENDOR:' HTTP Header with the OUI of the MAC Manager's Vendor in every HTTP query and response.
- The MAC Manager MAY populate an 'X-CL-VENDOR-STRING:' HTTP Header with a vendor-defined string representing the vendor name.
- The MAC Manager MUST populate an 'X-CL-SW-VERSION:' HTTP Header with the Software Version of the MAC Manager in every HTTP query and response. The exact format of the SW Version string is vendor-specific.

- The MAC Manager MUST populate an 'X-CL-HW-IDENTIFIER:' HTTP Header with a MAC Manager unique identifier defined as UUID version 4 as described in [RFC 4122].
- The MAC Manager MUST encode the UUIDv4 as base64 [RFC 4648].
- The MAC Manager MUST use the unique identifier sent in 'X-CL-HW-IDENTIFIER:' HTTP Header during the entire operation time.

7.2.5 Request URI

The MAC-NE MUST implement request URI handling as defined in [RFC 8040].

The MAC Manager MUST provide the request URI as defined in [RFC 7231] and [RFC 8040].

A MAC-NE expects the first line of an HTTP request from the MAC Manager to be of the form:

<op>/<base uri>/<path>?<query>

Where:

- <op> is the HTTP operation (GET, POST, PUT, etc.)
- <base uri> detailed below
- <path> is the RESTCONF path selector for a resource
- <query> is a RESTCONF query parameter

7.2.5.1 Base URI {+restconf}

The Base URI is determined through mechanisms specified in the RESTCONF RFC [RFC 8040]. This endpoint may be dynamic.

The MAC-NE MUST support Base URI detection through the '.well-known/host-meta' endpoint detailed in RESTCONF.

The MAC-NE MUST NOT change its Base URI while operational. The MAC-NE MAY change its Base URI on system boot.

The MAC Manager MUST support automatically determining the Base URI through the '.well-known/host-meta' endpoint when first connecting to a MAC-NE after the MAC-NE has booted.

Throughout this document and the MMI YANG modules, the Base URI is represented as {+restconf}, meaning that this part of the URI needs to be dynamically determined during runtime.

7.2.5.2 URI Parameters

The RESTCONF URI allows for variable parameters and identifies a resource. The resource hierarchy is encoded in a left-to-right manner, where a parent node is to the left of a child node. In this document variable parameters are represented as surrounded by '[' and ']'. For example: '[mac address]' represents a place on a URI that requires a concrete value to be provided by the MAC Manager.

The MAC-NE MUST follow the RESTCONF [RFC 8040] rules for reading and interpreting Resource Identifiers.

The MAC Manager MUST follow the RESTCONF [RFC 8040] rules for creating and interpreting Resource Identifiers.

7.2.6 HTTP Status Codes

The MAC-NE MUST use HTTP status codes to signal success and failure conditions compliant with [RFC 7231] and [RFC 8040]. Status codes are as follows:

- 2xx status codes are success cases
- 3xx status codes indicate action required to retry

- 4xx indicates error in request format or payload
- 5xx indicates error in the server.

At a minimum, the MAC-NE MUST provide HTTP status codes compliant with the list provided in Table 3 - HTTP Status Codes.

Table 3 - HTTP Status Codes

Status Code	Use
200 Ok	Operation was successful.
201 Created	Request fulfilled and a new resource was created.
301 Moved Permanently	Resource was permanently moved to the location returned in the 'Location:' HTTP Header.
400 Bad Request	Error status code when a request is bad or malformed.
401 Unauthorized	Error status code when an access was denied due to a security violation.
403 Forbidden	Error status code when access will never be provided.
404 Not Found	Error status code when access was attempted to an invalid or unknown URI.
406 Not Acceptable	Error status code with the MAC-NE cannot accept or provide a content encoding type.
412 Precondition Failed	Error status code when a MAC Manager attempts to update or change a MAC-NE resource and the ETAG values do not match.
428 Precondition Required	Error status code when a MAC Manager attempts to update or change a MAC-NE resource and did not provide an 'If-Match' header.
500 Internal Server Error	Error status code when MAC-NE has an internal error or fault.
501 Not Implemented	Error status code when MAC-NE has not implemented the resource at the provided URI.

The MAC-NE MAY provide other HTTP error codes compliant with [RFC 7231] and [RFC 8040].

The MAC-NE MUST provide an HTTP payload containing an error message compliant with the 'ietf-restconf:errors' YANG data model defined in [RFC 8040] for all 4xx and 5xx status codes.

The MAC Manager MUST provide an HTTP payload containing an error message compliant with the 'ietf-restconf:errors' YANG data model defined in [RFC 8040] for all 4xx and 5xx status codes.

7.2.7 RESTCONF Error Reporting

RESTCONF commands can fail at the protocol level when executing standard HTTP GET/PUT/POST commands or can fail during a RESTCONF RPC. This section discusses error conditions returned at the protocol level and would be returned when the HTTP action returns an HTTP error. When invoking an RPC that does not fail at the protocol level, the RPC error status information is carried in the 'output { }' subcontainer of the RPC.

The MAC-NE MUST support RESTCONF protocol error reporting per section 7.1 of [RFC 8040]. Protocol error responses are provided for HTTP status code classes “4xx” and “5xx”. [RFC 8040] provides a mapping from NETCONF <error-tag> to an HTTP status code.

When reporting an error message with a protocol error response, the MAC-NE MUST include an 'error-app-tag' string with the format of: "[{originator}-{error-tag}] {error-message}"; where {originator} is 1 for a vendor-defined error or 2 for a CableLabs. The CableLabs error-tag are defined as:

- internal-error
- unable-to-comply
- communication-loss
- invalid-input
- not-implemented
- duplicate

- entity-not-found
- object-in-use
- capacity-exceeded
- not-in-valid-state
- access-denied

When the MAC-NE is reporting an 'error-app-tag', an {error-type} of 1 indicates the {error-code} and {error-string} are vendor-defined. When the MAC-NE is reporting an 'error-app-tag', an {error-type} of 2 indicates the {error-code} and {error-string} are defined by CableLabs as above.

When a MAC-NE is reporting an 'error-app-tag', it MUST favor returning a CableLabs error defined in here, if possible before choosing to return a vendor-defined error.

7.2.8 RESTCONF Capabilities - Resource Discovery

The MAC-NE MUST implement the '{+restconf}/yang-library-version' URI defined in [RFC 8040].

The MAC-NE MUST implement the ietf-yang-library module defined in [RFC 7895].

The MAC-NE MUST populate the ietf-yang-library with the modules and submodules that the MAC-NE supports.

The MAC Manager MAY use the ietf-yang-library to determine which modules and submodules are supported by the MAC-NE.

7.2.9 RESTCONF Notifications

The MAC-NE MUST, as a RESTCONF server, implement RESTCONF notifications as per [RFC 8040] Section 6. This strengthens the requirement of [RFC 8040] Section 6.1 from a may to a must; all other may interpretations of [RFC 8040] Section 6 remain as-defined.

7.3 Optimistic Concurrency Control

In some architectures, a single MAC-NE device can have multiple MAC Managers communicating with it at the same time. In these cases, the MAC-NE makes use of Optimistic Concurrency Control (OCC) to manage multiple access rather than an explicit locking scheme. Optimistic Concurrency Control assumes that multiple writes rarely conflict and that most concurrency is done on the read level. This is congruent with the expected deployment architecture of FMA. Optimistic Concurrency Control is simple for the MAC-NE to implement and places the conflict resolution algorithms in the MAC Manager. An example of Optimistic Concurrency is provided in the following Figure 3.

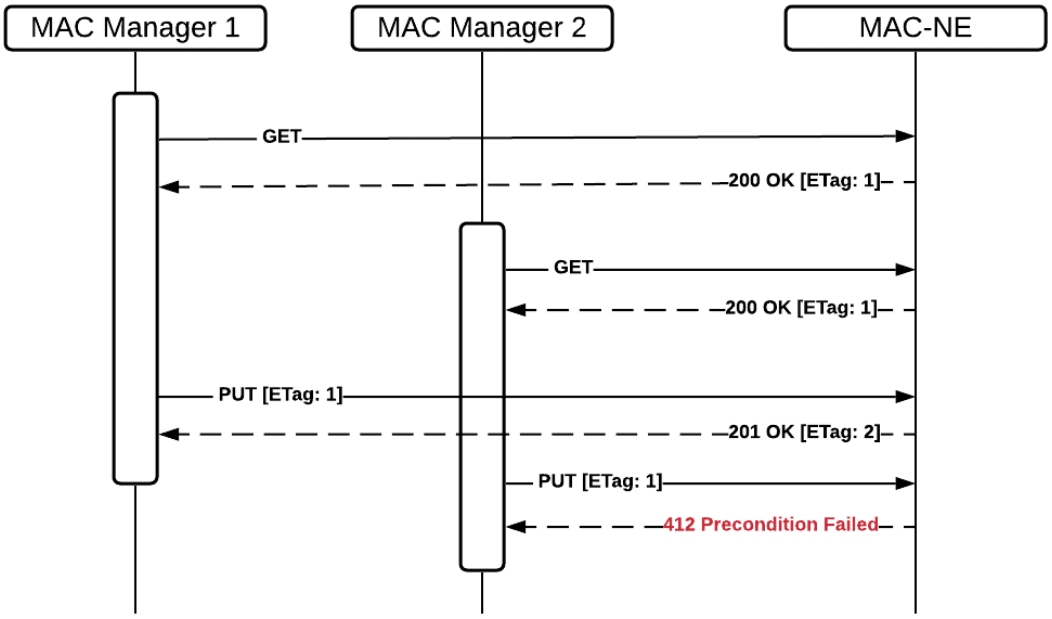

Figure 3 - Optimistic Concurrency Control Example

In this example, two MAC Managers are querying the MAC-NE for a shared resource. The MAC-NE reports an ETAG of "1", indicating a unique value for the current version of the resource. This example uses version-like semantics but that is not required for OCC to function, OCC can function with any ETAG semantics that guarantees the ETAG is unique and changes when the resource changes. Now, both MAC Managers have a copy of the resource with an ETAG of "1", next each MAC Manager independently changes the resource. MAC Manager 1 writes its new resource first including the ETAG of "1", the MAC-NE accepts the write and replies with a new ETAG of "2". Then MAC Manager 2 writes its new resource with the ETAG of "1" and the MAC-NE fails the write operation with a 412 status code. The MAC-NE fails the write because its internal ETAG for the resource is "2", while MAC Manager 2 included the old ETAG version "1" which indicates that MAC Manager 2 was operating on stale data. If the MAC Manager still wishes to attempt the write, it needs to read the resource again (getting ETAG "2"), modify the new resource values, and then attempt the write again with ETAG "2".

The MAC-NE MUST use Optimistic Concurrency Control on its RESTCONF interface to resolve write conflicts.

The MAC-NE MUST generate an HTTP ETAG header for all resource URIs.

The MAC-NE MUST generate a new HTTP ETAG whenever a resource changes. The MAC-NE MUST guarantee that each generated ETAG for a specific resource is unique. The algorithm used to generate unique ETAGs is left as an implementation detail and does not impact the operation of Optimistic Concurrency Control assuming they are unique each time the resource changes.

The MAC-NE MUST apply all changes in a single RESTCONF write in a transactional manner. The MAC-NE MUST include the generated ETAG for a resource in the HTTP header of every RESTCONF read.

The MAC Manager MUST include the IF-MATCH HTTP header in every RESTCONF write that is modifying an existing resource. When the MAC Manager does a RESTCONF write on an existing resource, the MAC Manager MUST populate the IF-MATCH value with the last ETAG the MAC Manager read for the resource being written. If the ETAG the MAC Manager provided does not match the current ETAG the MAC-NE calculated, the write will fail with error code '*412 Precondition Failed*', handling write conflicts within the MAC Manager is left as an implementation detail.

The MAC Manager design and architecture for redundancy could make use of the MAC-NE's Optimistic Concurrency Control features, but no specific design is mandated by this specification and high-availability of the MAC Manager is left for future study. MAC Manager conflict resolution algorithms are left as vendor-specific details.

8 SECURITY

Because of the disaggregated nature of distributed architectures such as FMA, the subcomponents that comprise the system will typically not be collocated in a secure facility. As a result, at least some of the components are expected to be deployed in insecure locations, whether that be on the pole, in a pedestal, or inside of a curb vault. For this reason, it is imperative that individual components be secured and that the links between system components be secured in order to insure the authenticity and integrity of the system.

The general approach to network security as it applies to the MMI interface includes the following:

- Use of X.509 certificates for authentication of system interfaces,
- Handshake protocols such as TLS providing methods for mutual authentication and negotiation of cryptographic methods to provide for encrypted communication between elements,
- Encryption of communication between authenticated networking components.

Other considerations and best practices can also be applied, although these details are outside of the scope of standardization. For example:

- Secured storage of secrets or private keys associated with certificates.
- Application of good design principles and security measures to remove or otherwise minimize the impact of attacks such as denial of service attacks, backdoor entries, operating system and application layer attacks, improper authorizations, password attacks or other common methods employed during cyber-attacks.

8.1 HTTPS Transport

The MAC-NE MUST implement support for HTTP1.1 [RFC 7231] over TLS1.2 [RFC 5246]. This protocol combination is commonly referred to as HTTPS.

The MAC-NE MAY also support HTTP1.1 over TLS1.3 [RFC 8446].

The MAC Manager MUST implement support for HTTP1.1 [RFC 7231] over TLS1.2 [RFC 5246].

The MAC Manager MAY also support HTTP1.1 over TLS1.3 [RFC 8446].

Further TLS requirements (such as path verification) are specified in [FMA-SYS].

The MAC-NE MUST reject any HTTP request that is not secured with TLS. The following optional requirements can also be implemented at the vendor's discretion:

- The MAC-NE SHOULD provide code 301 Moved Permanently and Location header in its request response when the HTTPS server is on the same MAC-NE and TLS was not used to secure the HTTP request.
- The MAC-NE MAY provide a status code of 403 Forbidden in its request response when the HTTP request is not secured via TLS.

The MAC-NE MUST support HTTP over TLS on TCP Port 443.

The MAC Manager MUST support HTTP over TLS on TCP Port 443.

Appendix I Acknowledgements

We wish to thank the following participants contributing directly to this document:

FMA Leadership Participants

At the time this specification was submitted to CableLabs for approval, the Flexible MAC Architecture (FMA) Working Group (WG) and Task Forces (TF) under the WG was organized with the following leadership roles and affiliations:

Flexible MAC Architecture (FMA) Working Group (WG)

FMA WG Co-Chair	Michael "Mike" Emmendorfer, CommScope Inc.
FMA WG Co-Chair	Jon Schnoor, Cable Television Laboratories, Inc.
FMA WG Co-Chair	Jeff Finkelstein, Cox Communications

Task Force 1: MAC Manager Interface

Task Force 1 Co-Chair	Douglas Johnson, Vecima Networks Inc.
Task Force 1 Co-Chair	Stephen Kraiman, CommScope Inc.

Task Force 2: PacketCable and Lawful Intercept

Task Force 2 Co-Chair	Rex Coldren, Vecima Networks Inc.
Task Force 2 Co-Chair	Dan Torbet, CommScope Inc.

Task Force 3: FMA OSS Interface Specification

Task Force 3 Co-Chair	Steve Burroughs, Cable Television Laboratories, Inc.
Task Force 3 Co-Chair	Dwain Frieh, CommScope Inc.

Task Force 4: Data Forwarding

Task Force 4 Co-Chair	Mike Patrick, Harmonic Inc.
Task Force 4 Co-Chair	Mircea Orban, CommScope Inc.

Task Force 5: FMA-Proactive Network Maintenance

Task Force 5 Co-Chair	Jason Rupe, Cable Television Laboratories, Inc.
Task Force 5 Co-Chair	Santhana Chari, CommScope Inc.

We wish to thank the following participants contributing directly to this document.

Contributor	Company Affiliation
Kirk Erichsen, Steve Goeringer, Brian Hedstrom, Jon Schnoor	CableLabs
Hongbiao Zhang	Casa Systems
Chris Bush	CommScope
Mike Emmendorfer	CommScope
Steve Foley	CommScope
Stephen Kraiman	CommScope
Tadeusz Ciesielski	Falcon V Systems
Andriy Korud	Falcon V Systems
Michael Patrick	Harmonic
Jorge Gil	Hirschmann Digital Access
Sujeeth Dharanikota	Nokia
Rex Coldren	Vecima
Douglas Johnson	Vecima
Utku Yilmaz	Vecima

Appendix II Revision History

The following Engineering Change was incorporated into CM-SP-FMA-MMI-I02-210526.

ECN Identifier	Accepted Date	Title of EC	Author
FMA-MMI-N-21.2160-1	4/28/2021	FMA-MMI I02 Candidate	Schnoor

The following Engineering Change was incorporated into CM-SP-FMA-MMI-I03-220126.

ECN Identifier	Accepted Date	Title of EC	Author
FMA-MMI-N-21.2217-2	1/6/2022	FMA MMI Spec I03 candidate	Schnoor

* * *