Technical Report

TR-383

Common YANG Modules for Access Networks

Issue: 01 Amendment 5

Issue Date: March 2022

Notice

The Broadband Forum is a non-profit corporation organized to create guidelines for broadband network system development and deployment. This Technical Report has been approved by members of the Forum. This Technical Report is subject to change. This Technical Report is owned and copyrighted by the Broadband Forum, and all rights are reserved. Portions of this Technical Report may be owned and/or copyrighted by Broadband Forum members.

Intellectual Property

Recipients of this Technical Report are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of this Technical Report, or use of any software code normatively referenced in this Technical Report, and to provide supporting documentation.

Terms of Use

1. License

Broadband Forum hereby grants you the right, without charge, on a perpetual, non-exclusive and worldwide basis, to utilize the Technical Report for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, products complying with the Technical Report, in all cases subject to the conditions set forth in this notice and any relevant patent and other intellectual property rights of third parties (which may include members of Broadband Forum). This license grant does not include the right to sublicense, modify or create derivative works based upon the Technical Report except to the extent this Technical Report includes text implementable in computer code, in which case your right under this License to create and modify derivative works is limited to modifying and creating derivative works of such code. For the avoidance of doubt, except as qualified by the preceding sentence, products implementing this Technical Report are not deemed to be derivative works of the Technical Report.

2. NO WARRANTIES

THIS TECHNICAL REPORT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT AND ANY IMPLIED WARRANTIES ARE EXPRESSLY DISCLAIMED. ANY USE OF THIS TECHNICAL REPORT SHALL BE MADE ENTIRELY AT THE USER'S OR IMPLEMENTER'S OWN RISK, AND NEITHER THE BROADBAND FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY USER, IMPLEMENTER, OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS TECHNICAL REPORT, INCLUDING BUT NOT LIMITED TO, ANY CONSEQUENTIAL, SPECIAL, PUNITIVE, INCIDENTAL, AND INDIRECT DAMAGES.

3. THIRD PARTY RIGHTS

Without limiting the generality of Section 2 above, BROADBAND FORUM ASSUMES NO RESPONSIBILITY TO COMPILE, CONFIRM, UPDATE OR MAKE PUBLIC ANY THIRD PARTY ASSERTIONS OF PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS THAT MIGHT NOW OR IN THE FUTURE BE INFRINGED BY AN IMPLEMENTATION OF THE TECHNICAL REPORT IN ITS CURRENT, OR IN ANY FUTURE FORM. IF ANY SUCH RIGHTS ARE DESCRIBED ON THE TECHNICAL REPORT, BROADBAND FORUM TAKES NO POSITION AS TO THE VALIDITY OR INVALIDITY OF SUCH ASSERTIONS, OR THAT ALL SUCH ASSERTIONS THAT HAVE OR MAY BE MADE ARE SO LISTED.

All copies of this Technical Report (or any portion hereof) must include the notices, legends, and other provisions set forth on this page.

Issue History

Issue Number	Approval Date	Publication Date	Issue Editor	Changes
1	8 May 2017	2 June 2017	Joey Boyd, ADTRAN Ludwig Pauwels, Nokia	Original
1 Amendment 1	13 June 2018	17 July 2018	Joey Boyd, ADTRAN Ludwig Pauwels, Nokia	Provide YANG model updates for Layer 2 Forwarding and QoS; publish initial model for Layer 2 Multicast Management; remove YANG models with dependencies on a draft revision of ietf-hardware.
1 Amendment 2	3 December 2018	3 December 2018	Joey Boyd, ADTRAN Ludwig Pauwels, Nokia	Add 'ethernet-like' abstract interface type.
1 Amendment 3	13 October 2020	13 October 2020	Nick Hancock, ADTRAN Ludwig Pauwels, Nokia	Enhancements to existing models. New models for ANCP and Hardware Management.
1 Amendment 4	2 June 2021	2 June 2021	Nick Hancock, ADTRAN Ludwig Pauwels, Nokia	Enhancements to existing models. New modules added to common, equipment, QoS, sub-interfaces and types.
1 Amendment 5	1 March 2022	1 March 2022	Nick Hancock, ADTRAN Ludwig Pauwels, Nokia	Enhancements to existing models. New models for Device Aggregation, Connectivity Fault Management (CFM) and Software Management. New modules added to ANCP and interfaces.

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

Editors:	Nick Hancock, ADTRAN Ludwig Pauwels, Nokia
Common YANG Work Area Directors:	Joey Boyd, ADTRAN Sven Ooghe, Nokia
YANG Modules Project Stream Leaders:	Joey Boyd, ADTRAN

Table of Contents

Table of Figures

Table of Tables

Executive Summary

This Technical Report defines YANG data models for the management of Broadband-Forum-specified access network equipment and network functions used across many deployment scenarios. There is no assumption for Broadband Forum YANG modules to apply outside the scope of Broadband Forum requirements.

The models specified in this Technical Report are independent of any management protocol, such as RESTCONF and NETCONF.

Amendment 5 to Issue 1 of this Technical Report:

- adds initial revisions of the following YANG modules:
  - bbf-ancp-alarm-types
  - bbf-device-aggregation
  - bbf-dot1q-cfm
  - bbf-dot1q-cfm-alarm-types
  - bbf-dot1q-cfm-interfaces
  - bbf-dot1q-cfm-interfaces-state
  - bbf-dot1q-cfm-l2-forwarding
  - bbf-interfaces-remote-hardware-state
  - bbf-software-management
  - bbf-software-management-voice;
- modifies the following YANG modules:
  - bbf-ancp:
    - adds a configurable administrative state ‘enable’ to a session
    - adds an operational state for the transport connection to the Network Access Server (NAS)
    - adds states ‘pending’ and ‘disabled’ to the adjacency state of a session
    - adds a container and child nodes that provide additional information regarding a malfunction encountered with the Adjacency Protocol
    - clarifies the behavior of the ‘use-leading-zeroes’ within the access line identification configuration
    - adds an action ‘reset’ to reset sessions that may have been halted due to a malfunction
  - bbf-ancp-interfaces:
    - adds statistics for topology discovery messages sent per access line to a session
  - bbf-frame-classification:
    - adds grouping ‘dscp-match-specific’
  - bbf-frame-processing-profiles:
    - extends the match criteria in the frame-processing-profile to match on ‘any frame’
    - extends the match criteria in the frame-processing-profile to match on a range of DSCP values
    - extends the exclude criteria in the frame-processing-profile to exclude a range of DSCP values

        - extends the ingress rewrite options in the frame-processing-profile to define a list of VLAN tags to push
        - extends the egress rewrite options in the frame-processing-profile to define the number of VLAN tags to pop
    - bbf-hardware:
        - corrects the error message of the must statement on the leaf-list ‘port-layer-if
    - bbf-hardware-transceivers
        - deprecates the enum value ‘1000-mbytes-sec’ and adds the additional enum value ‘1600-mbytes-sec’ to the enumeration of the leaf ‘fibre-channel-speed’
        - changes type of the leafs ‘tx-bias-high-alarm’, ‘tx-bias-low-alarm’, ‘tx-bias-high-warning’ and ‘tx-bias-low-warning’ from uint16 to uint32
    - bbf-hardware-types:
        - adds identities to represent separate hardware that is not part of a device's own hardware
    - bbf-l2-dhcpv4-relay:
        - clarifies the behavior of the ‘use-leading-zeroes’ within the option 82 format configuration
    - bbf-ldra:
        - clarifies the behavior of the ‘use-leading-zeroes’ within the relay options configuration
    - bbf-interfaces-performance-management:
        - adds a timestamp to the performance history intervals
    - bbf-interfaces-statistics-management:
        - adds configuration associated with statistics related to an interface
    - bbf-pppoe-intermediate-agent:
        - clarifies the behavior of the ‘use-leading-zeroes’ within the vendor-specific tag configuration
    - bbf-qos-enhanced-scheduling:
        - removes the when statement on the augmentation of the interface and makes the container ‘egress-tm-objects’ a presence container
    - bbf-qos-policing
        - unused feature ‘policing-profile-ref’ removed
    - bbf-qos-traffic-mngt
        - adds a presence statement to the container ‘tm-root’, because a VLAN sub-interface does not have queues or a scheduling hierarchy
    - bbf-sub-interface-tagging:
        - extends the match criteria of the flexible match for in-line frame processing on a VLAN sub-interface to match on a range of DSCP values
        - extends the exclude criteria of the flexible match for in-line frame processing on an VLAN sub-interface to exclude a range of DSCP values
        - removes feature ‘copy-vlan-id-from-tag-index’ from the enum of the enumeration for the leaf ‘vlan-id’ within the container ‘dot1q-tag’;

- makes editorial changes only to the following YANG modules:
    - bbf-ancp-fastdsl-access-extensions
    - bbf-ancp-fastdsl-threshold
    - bbf-availability
    - bbf-device
    - bbf-hardware-cpu
    - bbf-interfaces-performance-management
    - bbf-interface-usage
    - bbf-l2-dhcpv4-relay-forwarding
    - bbf-l2-forwarding
    - bbf-l2-terminations
    - bbf-location
    - bbf-mgmd
    - bbf-mgmd-mrd
    - bbf-mgmd-types
    - bbf-network-types
    - bbf-qos-classifiers
    - bbf-qos-composite-filters
    - bbf-qos-policies
    - bbf-qos-policies-sub-interface-rewrite
    - bbf-qos-rate-control
    - bbf-subscriber-profiles
    - bbf-subscriber-types
    - bbf-yang-types;
- adds the following new sections to the Technical Report:
    - Section 7.4 Layer 2 Termination Interfaces
    - Section 11 Software Management;
- moves contents of section 7.1.1 to section 7.3 and expands the section to make it clearer as to which interfaces Interface Usage applies to;
- makes editorial changes to:
    - Section 6 Dependencies on related YANG modules and Standards.

1 Purpose and Scope

1.1 Purpose

This Technical Report defines YANG data models for the management of Broadband-Forum-specified access network equipment used across many deployment scenarios. Broadband-Forum-specified access network equipment comprises Access Nodes and FTTdp DPUs. There is no assumption for BBF YANG modules to apply globally, e.g., to apply to access network equipment other than BBF Access Nodes and FTTdp DPUs, or to apply to core network equipment.

The models specified in this Technical Report are independent of any management protocol.

1.2 Scope

The data models defined by this Technical Report support the Broadband Forum requirements as applicable to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs) and form the set of core models which can be used for a multitude of other applications. It is intended that data models which are application specific can be built on, reference, and/or function alongside the common models.

The figure below provides a high level view of the functionality covered by this Technical Report (BBF YANG in green):

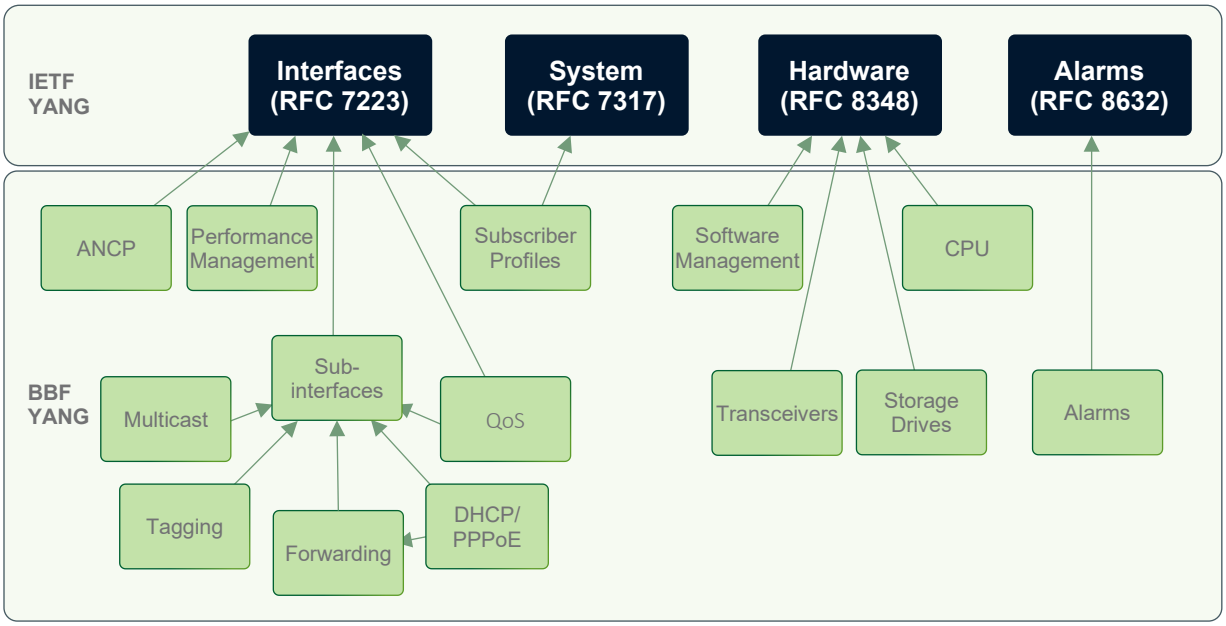

Figure 1 – YANG Data Model Relationships

2 References and Terminology

2.1 Conventions

In this Technical Report, several words are used to signify the requirements of the specification. These words are always capitalized. More information can be found be in RFC 2119 [10].

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

2.2 References

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

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

Document		Title	Source	Year
[1]	TR-101i2	Migration to Ethernet-Based Broadband Aggregation	BBF	2011
[2]	TR-147	Layer 2 Control Mechanism for Broadband Multi-Service Architectures	BBF	2008
[3]	TR-178	Multi-service Broadband Network Architecture and Nodal Requirements	BBF	2014
[4]	TR-178i2	Multi-service Broadband Network Architecture and Nodal Requirements	BBF	2017
[5]	TR-301i2a1	Architecture and Requirements for Fiber to the Distribution Point	BBF	2020
[6]	TR-355	YANG Modules for FTTdp Management	BBF	2020

Document		Title	Source	Year
[7]	IEEE 802.1Q	Bridges and Bridged Networks	IEEE	2018
[8]	IEEE 802.1Qcx	YANG Data Model for Connectivity Fault Management	IEEE	2020
[9]	IEEE 802.3	Ethernet Specification	IEEE	2015
[10]	RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	IETF	1997
[11]	RFC 2710	Multicast Listener Discovery (MLD) for IPv6	IETF	1999
[12]	RFC 2790	Host Resources MIB	IETF	2000
[13]	RFC 2863	The Interfaces Group MIB	IETF	2000
[14]	RFC 2933	Internet Group Management Protocol MIB	IETF	1999
[15]	RFC 3046	DHCP Relay Agent Information Option	IETF	2001
[16]	RFC 3315	Dynamic Host Configuration Protocol for IPv6 (DHCPv6)	IETF	2003
[17]	RFC 3376	Internet Group Management Protocol, Version 3	IETF	2002
[18]	RFC 3590	Source Address Selection for the Multicast Listener Discovery (MLD) Protocol	IETF	2003
[19]	RFC 3810	Multicast Listener Discovery Version 2 (MLDv2) for IPv6	IETF	2004
[20]	RFC 4119	A Presence-based GEOPRIV Location Object Format	IETF	2005
[21]	RFC 4243	Vendor-Specific Information Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option	IETF	2005
[22]	RFC 4286	Multicast Router Discovery	IETF	2005
[23]	RFC 4541	Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches	IETF	2006
[24]	RFC 4580	Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Subscriber-ID Option	IETF	2006
[25]	RFC 4605	Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding (“IGMP/MLD Proxying”)	IETF	2006
[26]	RFC 4649	Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option	IETF	2006
[27]	RFC 5519	Multicast Group Membership Discovery MIB	IETF	2009
[28]	RFC 5851	Framework and Requirements for an Access Node Control Mechanism in Broadband Multi-Service Networks	IETF	2010
[29]	RFC 6221	Lightweight DHCPv6 Relay Agent	IETF	2011
[30]	RFC 6320	Protocol for Access Node Control Mechanism in Broadband Networks	IETF	2011
[31]	RFC 6933	Entity MIB (Version 4)	IETF	2013
[32]	RFC 6991	Common YANG Data Types	IETF	2013
[33]	RFC 7223	A YANG Data Model for Interface Management	IETF	2014
[34]	RFC 7317	A YANG Data Model for System Management	IETF	2014
[35]	RFC 7950	The YANG 1.1 Data Modeling Language	IETF	2016

Document		Title	Source	Year
[36]	RFC 8342	Network Management Datastore Architecture (NMDA)	IETF	2018
[37]	RFC 8348	A YANG Data Model for Hardware Management	IETF	2018
[38]	RFC 8415	Dynamic Host Configuration Protocol for IPv6 (DHCPv6)	IETF	2018
[39]	RFC 8519	YANG Data Model for Network Access Control Lists (ACLs)	IETF	2019
[40]	RFC 8528	YANG Schema Mount	IETF	2019
[41]	RFC 8632	A YANG Data Model for Alarm Management	IETF	2019
[42]	G.988	ONU management and control interface (OMCI) specification	ITU-T	2017
[43]	G.997.1	Physical layer management for digital subscriber line transceivers	ITU-T	2019
[44]	G.997.2	Physical layer management for G.fast transceivers	ITU-T	2019
[45]	X.731	Information Technology – Open Systems Interconnection – Systems Management: State management function	ITU-T	1992
[46]	X.733	Information technology – Open Systems Interconnection – Systems Management: Alarm reporting function	ITU-T	1992
[47]	X.736	Information technology – Open Systems Interconnection – Systems Management: Security alarm reporting function	ITU-T	1992
[48]	MEF 10.2	Ethernet Services Attributes Phase 2	MEF	2009
[49]	MEF 10.3	Ethernet Services Attributes Phase 3	MEF	2013
[50]	TR 32.859	Telecommunication management; Study on Alarm Management	3GPP	2013
[51]	TS 32.111-2	Telecommunication management; Fault Management; Part 2: Alarm Integration Reference Point (IRP): Information Service (IS) (Release 14)	3GPP	2017

2.3 Definitions

The following terminology is used throughout this Technical Report.

Model	A data model.
Module	A YANG module defines the hierarchy of data for the data model.
Submodule	A YANG module may be broken up into multiple submodules for ease of maintainability. The overall data model is comprised of a module and zero or more submodules.

2.4 Abbreviations

This Technical Report uses the following abbreviations:

AN	Access Node
ANCP	Access Node Control Protocol
CFM	Connectivity Fault Management
DEI	Drop Eligible Indicator
DHCP	Dynamic Host Configuration Protocol
DPU	Distribution Point Unit
DSL	Digital Subscriber Line
FastDSL	DSL or G.fast
FRA	Fast Rate Adaptation
FTTdp	Fiber to the Distribution Point
FTU	FAST Transceiver Unit
IGMP	Internet Group Management Protocol
L2	Layer 2
LAG	Link Aggregation Group
MGMD	Multicast Group Membership Discovery
MLD	Multicast Listener Discovery
NAS	Network Access Server
NMDA	Network Management Datastore Architecture
NT	Network Termination
PMA	Persistent Management Agent
PPPoE	Point-to-Point Protocol over Ethernet
RPC	Remote Procedure Call
SRA	Seamless Rate Adaptation
TLV	Type-Length-Value
TR	Technical Report
UML	Unified Modeling Language™
URL	Uniform Resource Locator
WA	Work Area
WT	Working Text

3 Technical Report Impact

3.1 Energy Efficiency

TR-383 has no impact on energy efficiency.

3.2 IPv6

TR-383 includes YANG modules that support IPv6 deployments.

3.3 Security

TR-383 has no impact on security.

3.4 Privacy

Any issues regarding privacy are not affected by TR-383.

4 Modules

The YANG modules contained in TR-383 are briefly described here. These modules are published on GitHub at https://github.com/BroadbandForum/yang/.

4.1 DHCP

These modules provide functionality to manage DHCP and can be found in the *networking* directory on GitHub.

4.1.1 bbf-l2-dhcpv4-relay

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on subscriber management via the DHCPv4 protocol as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

4.1.2 bbf-l2-dhcpv4-relay-forwarding

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on subscriber management via the DHCPv4 protocol as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments bbf-l2-forwarding with subscriber management via the DHCPv4 protocol [15].

4.1.3 bbf-ldra

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on subscriber management via the DHCPv6 protocol as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

This functionality is also known as a Lightweight DHCPv6 Relay Agent (LDRA) [29].

4.2 Equipment

These modules provide management extensions related to hardware components as defined in the IETF RFC 8348 [37] and can be found in the *equipment* directory on GitHub.

4.2.1 bbf-hardware

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on hardware and interface management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the ietf-hardware model with additional management common to multiple classes of hardware components and augments the ietf-interfaces model to enable interfaces to reference hardware components.

4.2.2 bbf-hardware-cpu

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on hardware management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the ietf-hardware model with the management of a CPU processor (with single or multiple cores).

4.2.3 bbf-hardware-storage-drives

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on hardware management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the ietf-hardware model with the management of storage drives.

4.2.4 bbf-hardware-transceivers

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on hardware management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the ietf-hardware model with management of compact transceivers.

4.2.5 bbf-hardware-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on reusable data types as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module specializes types defined in the iana-hardware model.

4.2.6 bbf-hardware-transceiver-alarm-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on hardware management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a set of alarm definitions related to the management of compact transceivers.

4.3 Ethernet

These modules are specific to the management of Ethernet interfaces as defined by IEEE 802.3 [9] and can be found in the *interface* directory on GitHub.

4.3.1 bbf-ethernet-performance-management

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on Ethernet interface management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments bbf-interface-performance-management with Ethernet-specific counters.

4.4 Layer 2 Forwarding

These modules and their submodules are used for the management of Layer 2 (L2) Forwarding and can be found in the *networking* directory on GitHub.

4.4.1 bbf-l2-forwarding

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on layer 2 forwarding as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

4.4.1.1 bbf-l2-forwarding-base

This submodule contains a collection of YANG definitions for defining the top-level nodes for forwarding.

4.4.1.2 bbf-l2-forwarding-flooding-policies

This submodule contains a collection of YANG definitions for managing flooding policies.

Flooding policies define how the system forwards frames in case other forwarding mechanisms do not arrive at a forwarding decision.

4.4.1.3 bbf-l2-forwarding-forwarders

This submodule contains a collection of YANG definitions for managing forwarders.

A forwarder is used to forward traffic between two or more interfaces.

4.4.1.4 bbf-l2-forwarding-forwarding-databases

This submodule contains a collection of YANG definitions for managing forwarding databases.

A forwarding database contains the necessary information regarding the MAC addresses which are used in the forwarding decision.

4.4.1.5 bbf-l2-forwarding-mac-learning-control

This submodule contains a collection of YANG definitions for managing MAC address learning constraints, i.e., to constrain MAC learning rules compared with the standard IEEE MAC learning.

4.4.1.6 bbf-l2-forwarding-mac-learning

This submodule contains a collection of YANG definitions for managing MAC learning.

For a forwarder, it specifies the forwarding database to use for the specified forwarder. For an interface, it provides the ability to enable/disable MAC learning as well as specifies other parameters associated with MAC learning.

4.4.1.7 bbf-l2-forwarding-split-horizon-profiles

This submodule contains a collection of YANG definitions for managing split horizon profiles.

These profiles allow (or disallow) forwarding between various forwarder ports based on the underlying interface usage.

4.4.1.8 bbf-l2-forwarding-shared-fdb

Replaced by section 4.4.2.

4.4.2 bbf-l2-forwarding-shared-fdb

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on layer 2 forwarding as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of YANG definitions for managing shared forwarding databases.

4.5 Interfaces

These modules augment ietf-interfaces [33] with additional interface management and can be found in the *interface* directory on GitHub.

4.5.1 bbf-interfaces-performance-management

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on interface management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module reports performance management of statistics defined by the IETF interfaces data model, ietf-interfaces (RFC 7223) [33].

4.5.2 bbf-interfaces-statistics-management

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on interface management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments ietf-interfaces [33] with a reset action for statistics.

4.5.3 bbf-interface-usage

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on interface management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of YANG definitions defining how interfaces are used.

4.5.4 bbf-ptm

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on interface management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the IETF interfaces data model, ietf-interfaces (RFC 7223) [33], with nodes for managing Packet Transfer Mode (PTM) interfaces.

4.5.5 bbf-l2-terminations

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Layer 2 terminations as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

4.5.6 bbf-interfaces-remote-hardware-state

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on interface management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the IETF interfaces data model, ietf-interfaces (RFC 7223), with nodes for monitoring the software of remote hardware.

4.6 PPPoE

These modules provide functionality to manage Point-to-Point Protocol over Ethernet and can be found in the *networking* directory on GitHub.

4.6.1 bbf-pppoe-intermediate-agent

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on subscriber management via the PPPoE protocol as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified access Nodes and FTTdp DPUs).

4.7 QoS

These modules provide functionality to manage Quality of Service (QoS) and can be found in the *networking* directory on GitHub.

4.7.1 bbf-qos-classifiers

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of classifiers that can be used to classify frames and assign actions to be applied to those frames.

4.7.2 bbf-qos-filters

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains definitions of filter criteria.

4.7.3 bbf-qos-policies

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of policies that can be used to control the flow of packets.

4.7.4 bbf-qos-policies-sub-interfaces

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments sub-interfaces to support policies to control the flow of packets.

4.7.5 bbf-qos-rate-control

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments classifiers to control frame rates.

4.7.6 bbf-qos-traffic-mngt

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of functions for QoS traffic management (TM).

4.7.7 bbf-qos-enhanced-scheduling

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments interfaces to add additional configuration to manage enhanced traffic scheduling.

4.7.8 bbf-qos-policer-envelope-profiles

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments classifiers to add management of envelope policing as described in MEF 10.3 [49].

4.7.9 bbf-qos-policing-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains common types used for management of policers.

4.7.10 bbf-qos-policing

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments classifiers to manage the policing of flows.

4.7.11 bbf-qos-shaping

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments traffic management profiles with shaper profiles and augments interfaces to reference a shaper profile.

4.7.12 bbf-qos-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains type definitions used in multiple QoS modules.

4.7.13 bbf-qos-composite-filters

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains definitions of filter criteria for Ethernet header fields, IPv4 and IPv6 header fields, some IP packet payload fields, and it contains filters composed of a combination of these fields.

4.7.14 bbf-qos-policies-sub-interface-rewrite

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains augments to sub-interfaces to support policies applied to packets.

4.7.15 bbf-qos-traffic-mngt-state

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of functions for monitoring QoS traffic management (TM). This module is to be used, along with 'bbf-qos-traffic-mngt', when the server does not support Network Management Datastore Architecture (NMDA) as defined in RFC 8342 [36].

4.7.16 bbf-qos-enhanced-scheduling-state

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments interfaces to add monitoring of enhanced traffic scheduling. This module is to be used, along with 'bbf-qos-enhanced-scheduling', when the server does not support Network Management Datastore Architecture (NMDA) as defined in RFC 8342 [36].

4.7.17 bbf-qos-policies-state

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments interfaces to add monitoring of policies that can be used to control the flow of packets. This module is to be used, along with ‘bbf-qos-policies’, when the server does not support Network Management Datastore Architecture (NMDA) as defined in RFC 8342 [36].

4.7.18 bbf-qos-policing-state

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of Quality of Service (QoS) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments interfaces to add monitoring the policing of flows. This module is to be used, along with 'bbf-qos-policing', when the server does not support Network Management Datastore Architecture (NMDA) as defined in RFC 8342 [36].

4.8 Sub-interfaces

These modules provide management definitions for sub-interfaces and can be found in the *interface* directory on GitHub.

4.8.1 bbf-frame-classification

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on frame classification as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains reusable groupings defined for frame classification.

4.8.2 bbf-sub-interface-tagging

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of sub-interfaces as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the frame processing configuration of a (sub-)interface with additional criteria and adds VLAN-specific ingress and egress rewrite actions.

4.8.3 bbf-sub-interfaces

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of sub-interfaces as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments an interface with sub-interface-specific frame processing configuration.

4.8.4 bbf-frame-processing-profiles

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of sub-interfaces as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains the definition of a profile for classifying frames and performing ingress- and egress-rewrite actions, and it augments the frame processing configuration of a (VLAN sub-)interface for using this profile.

4.9 Subscribers

These modules provide management of subscriber-related functionality and can be found in the *networking* directory on GitHub.

4.9.1 bbf-subscriber-profiles

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of subscribers as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module adds support for subscriber profiles and augments an interface to enable an interface to a reference a subscriber profile. It also augments ietf-system to add system-specific subscriber management.

4.9.2 bbf-subscriber-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of subscribers as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module defines common types associated with subscribers and subscriber protocols.

4.10 Types

These modules provide reusable type definitions for use across all BBF YANG models and can be found in the *common* directory on GitHub unless otherwise specified.

4.10.1 bbf-dot1q-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on reusable data types as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module defines common types for support of IEEE 802.1Q [7].

4.10.2 bbf-if-type

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on reusable data types as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module defines interface types that are needed for BBF applications but are not defined in iana-if-type.

This module can be found in the *interface* directory on GitHub.

4.10.3 bbf-inet-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on reusable data types as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module defines additional YANG data types that are useful in managing Internet-Protocol-related configuration that are not defined by the IETF.

4.10.4 bbf-yang-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on reusable data types as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module defines common types used throughout BBF data models.

4.10.5 bbf-device-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on managing physical devices.

Specifically, this module defines common types associated with device management.

4.10.6 bbf-location-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on providing location information associated with network functions.

Specifically, this module defines common types associated with location information.

4.10.7 bbf-network-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on providing network information associated with network functions.

Specifically, this module defines common types associated with network information.

4.10.8 bbf-node-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on providing network node information associated with network functions.

Specifically, this module defines common types associated with network nodes.

4.11 Common

These modules provide support for common requirements for use across all BBF YANG models and can be found in the *common* directory on GitHub.

4.11.1 bbf-availability

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the general availability of specific resources as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

4.11.2 bbf-contact

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on providing contact information regarding network functions.

4.11.3 bbf-device

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on managing physical devices.

4.11.4 bbf-end-user

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on providing information for end users of network functions.

4.11.5 bbf-location

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on providing location information associated with network functions.

4.12 Layer 2 Multicast

These modules ant their submodules are used for the management of Layer 2 (L2) Multicast and can be found in the *networking* directory on GitHub.

4.12.1 bbf-mgmd

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on layer 2 multicast management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of YANG definitions for supporting the Multicast Group Membership Discovery (MGMD) Protocols. In particular it, describes data nodes used for managing the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol in systems that act as a multicast proxy, snooper, or a snooper with proxy reporting.

4.12.1.1 bbf-mgmd-configuration-interface-to-host

This submodule contains a collection of YANG definitions for supporting the Multicast Group Membership Discovery (MGMD) Protocols. In particular, it describes configuration data nodes used for managing the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol on interfaces that connect the system to multicast hosts.

4.12.1.2 bbf-mgmd-configuration-interface-to-router

This submodule contains a collection of YANG definitions for supporting the Multicast Group Membership Discovery (MGMD) Protocols. In particular, it describes configuration data nodes used for managing the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol on interfaces that connect to multicast routers.

4.12.1.3 bbf-mgmd-configuration-multicast-snoop

This submodule contains a collection of YANG definitions for supporting the Multicast Group Membership Discovery (MGMD) Protocols. In particular, it describes configuration data nodes used for managing the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol in case the system acts as a snooper.

4.12.1.4 bbf-mgmd-operational-interface-to-host

This submodule contains a collection of YANG definitions for supporting the Multicast Group Membership Discovery (MGMD) Protocols. In particular, it describes state data nodes used for managing the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol on interfaces that connect a system to multicast hosts.

4.12.1.5 bbf-mgmd-operational-interface-to-router

This submodule contains a collection of YANG definitions for supporting the Multicast Group Membership Discovery (MGMD) Protocols. In particular, it describes state data nodes used for managing the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol on interfaces that connect a system to multicast routers.

4.12.2 bbf-mgmd-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on L2 multicast management as applicable to access network equipment. This module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of YANG type and feature definitions for use in modules supporting the Multicast Group Membership Discovery (MGMD) Protocols. In particular, it describes data nodes used for managing the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol.

4.12.3 bbf-mgmd-mrd

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on layer 2 multicast management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a collection of YANG definitions for supporting the Multicast Group Membership Discovery (MGMD) Protocols. In particular, it describes data nodes used for managing the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol in systems that act as a multicast proxy, snooper, or a snooper with proxy reporting.

4.13 Alarms

These modules add BBF-specific alarm definitions based on ietf-alarms (RFC 8632 [41]) and can be found in the *common* directory on GitHub.

4.13.1 bbf-alarm-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the management of alarms as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module defines abstract alarm types that are needed for BBF applications to be able to define their own specific abstract and concrete alarm types.

4.14 ANCP

These modules provide management of the Access Node Control Protocol (ANCP) and can be found in the *networking* directory on GitHub.

4.14.1 bbf-ancp

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the Access Node Control Protocol (ANCP) as defined in RFC 6320 [30]. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

This data model is designed for the Network Management Datastore Architecture defined in RFC 8342 [36].

4.14.2 bbf-ancp-interfaces

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the Access Node Control Protocol (ANCP) as defined in RFC 6320 [30]. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments ietf-interfaces [33] to manage individual access lines that participate in ANCP.

4.14.3 bbf-ancp-fastdsl-access-extensions

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the Access Node Control Protocol (ANCP) as defined in RFC 6320 [30]. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments bbf-ancp to extend the definitions for FastDSL access technologies, which are used in the management of the Access Node side of the protocol.

4.14.4 bbf-ancp-fastdsl-threshold

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the Access Node Control Protocol (ANCP) as defined in RFC 6320 [30]. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments bbf-ancp to add the data nodes to manage line state reporting of the Access Node.

This data model is designed for the Network Management Datastore Architecture defined in RFC 6320 [30]. The line state reporting requirements are defined in BBF TR-147 [2].

4.14.5 bbf-ancp-alarm-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on the Access Node Control Protocol (ANCP) as defined in RFC 6320 [30]. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a set of alarm definitions.

4.15 Aggregation

This module supports the aggregation of devices and can be found in the *common* directory on GitHub.

4.15.1 bbf-device-aggregation

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on device aggregation. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

4.16 CFM

These modules bind the IEEE 802.1Qcx CFM YANG data model [8] to the Broadband Forum Layer 2 Forwarding and Alarm Management and can be found in the *networking* directory on GitHub.

4.16.1 bbf-dot1q-cfm

This module is part of a collection of YANG definitions for supporting the Broadband Forum requirements on Connectivity Fault Management (CFM) Operations, Administration, and Maintenance (OAM) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the IEEE 802.1Qcx CFM YANG data model [8] with data nodes to support BBF-specific requirements.

4.16.2 bbf-dot1q-cfm-alarm-types

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on Connectivity Fault Management (CFM) Operations, Administration, and Maintenance (OAM) as applicable to

access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module contains a set of alarm definitions.

4.16.3 bbf-dot1q-cfm-interfaces

This module is part of a collection of YANG definitions for supporting the Broadband Forum requirements on Connectivity Fault Management (CFM) Operations, Administration, and Maintenance (OAM) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the IEEE 802.1Qcx CFM YANG data model [8] to link a Maintenance Association (MA) End Point (MEP) to an interface.

4.16.4 bbf-dot1q-cfm-interface-state

This module is part of a collection of YANG definitions for supporting the Broadband Forum requirements on Connectivity Fault Management (CFM) Operations, Administration, and Maintenance (OAM) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the IEEE 802.1Qcx CFM YANG data model [8] to provide data nodes to display Maintenance Points present on an interface, i.e., the CFM stack of an interface.

4.16.5 bbf-dot1q-cfm-l2-forwarding

This module is part of a collection of YANG definitions for supporting the Broadband Forum requirements on Connectivity Fault Management (CFM) Operations, Administration, and Maintenance (OAM) as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

Specifically, this module augments the IEEE 802.1Qcx CFM YANG data model [8] to link a Maintenance Association (MA) to a forwarder.

4.17 Software

These modules provide management of software on hardware components and can be found in the *equipment* directory on GitHub.

4.17.1 bbf-software-management

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on software management as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

This module augments ietf-hardware.

4.17.2 bbf-software-management-voice

This module contains a collection of YANG definitions for supporting the Broadband Forum requirements on software management specific to voice applications as applicable to access network equipment. As such, this module is specific to access network equipment (e.g., BBF-specified Access Nodes and FTTdp DPUs).

5 Documentation

There are “README.md” files; these are short text files giving brief descriptions of the contents of the directories they are in.

Documentation for each module can be found in the *docs* folder of the corresponding directory, e.g., *networking*. For each top-level module, there is a *.tree file which provides a tree diagram of the module.

Additionally, in the *docs* folder under *common*, there are two files corresponding to the complete set of TR-383 YANG data models:

- bbf-common.tree: Provides a tree diagram comprised of all modules
- bbf-common.xml: Provides an XML schema representation of all modules

6 Dependencies on related YANG modules and Standards

TR-383 is based on YANG 1.1 (RFC 7950 [35]).

The following YANG modules are used by TR-383:

- Iana-hardware@2018-03-13
- iana-if-type@2021-06-21
- ieee802-dot1q-cfm@2020-06-04 [8]
- ieee802-dot1q-cfm-types@2020-06-04 [8]
- ieee802-dot1q-types@2020-06-04 [8]
- ieee802-types@2020-06-04 [8]
- ietf-alarms@2019-09-11 [41]
- ietf-alarms-x733@2019-09-11 [41]
- ietf-hardware@2018-03-13 [37]
- ietf-inet-types@2013-07-15 [32]
- ietf-interfaces@2014-05-08 [33]
- ietf-packet-fields@2019-03-04 [39]
- ietf-system@2014-08-06 [34]
- ietf-yang-schema-mount@2019-01-14 [40]
- ietf-yang-types@2013-07-15 [32]

7 Layer 2 Forwarding Data Model

The intent of this section is to provide some general information regarding the usage of the layer 2 forwarding data model. It is not possible to describe every possible application that would use the model, but rather it provides the theory behind the model and illustrates some general use cases.

7.1 Sub-interfaces

Before traffic can be forwarded, it must first be classified to determine what to forward, where to forward, and how to manipulate the packet if so desired. The concept of a VLAN sub-interface, realized in YANG as an interface of the type vlan-sub-interface, has been introduced for providing an interface which can be used as the source or destination interface of a forwarding decision. Each VLAN sub-interface classifies traffic from a particular lower layer interface into a forwarder. This classification consists of a set of rules specified using match criteria on to packet fields (e.g., VLAN-ID, p-bit). The lower layer interface can be either a non-aggregated physical or logical interface (e.g., Ethernet), an aggregation of physical or logical interfaces (e.g., LAG) or can be another VLAN sub-interface.

A VLAN sub-interface is created each time a new forwarding context is required (e.g., 1:1 VLAN). Each VLAN sub-interface can then have multiple rules associated with it if different classification results in the same forwarding decision. For example, one rule can catch frames tagged with a particular VLAN-ID, a second rule can catch untagged frames, and a third rule can catch priority-tagged frames. The second and third rules in this example cover the concept of a port default VLAN.

As stated above, multiple VLAN sub-interfaces can refer to the same lower layer interface in order to provide multiple traffic classifications based on different, but potentially overlapping, match criteria. In order to provide deterministic classification, each rule is given a priority. The scope of the priority is over all rules defined within all VLAN sub-interfaces referring to same lower layer interface. A packet ingressing the lower layer interface would then be compared to each rule starting with the highest priority rule and proceeding to the lowest priority rule. If a match occurs, the packet is process accordingly. If no match occurs, the packet is dropped.

The figure below shows how two VLAN sub-interfaces are associated with the same physical interface classifying traffic for two different forwarding decisions.

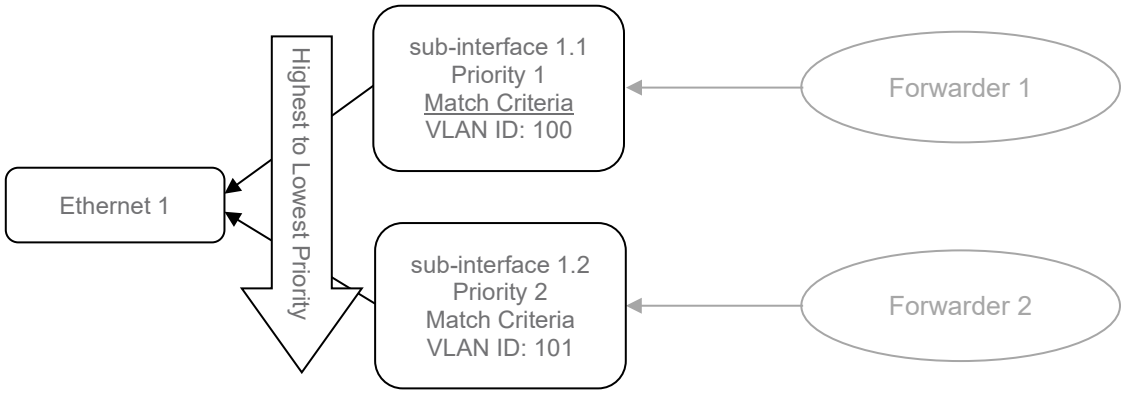

Figure 2 – Sub-interface Example

In addition to classification of traffic, the sub-interface also contains rules for any ingress or egress actions to take on each matched packet. These actions include pushing or popping tags, rewrite of p-bits, or rewrite of Drop Eligible Indication (DEI) bits.

7.1.1 Interface Usage

Replaced by section 7.3.

7.2 Forwarders

Once traffic has been classified and possibly manipulated, it needs to be forwarded appropriately to another sub-interface. A forwarder is used to determine how traffic is routed between two or more forwarder ports, each of which is associated with a sub-interface. This forwarder can be used to handle 1:1 VLAN, N:1 VLAN and N:M VLAN applications.

7.2.1 Forwarder Ports and Port Groups

Each forwarder port is associated with a sub-interface whose underlying interface is either a user port, a network port, or a subtended node port. Forwarder ports with similar forwarding characteristics can be placed into forwarding groups and referenced collectively when configuring the forwarder.

Figure 3 below shows the relationships between a forwarder, its forwarder ports, and the referenced sub-interfaces and their lower layer interfaces.

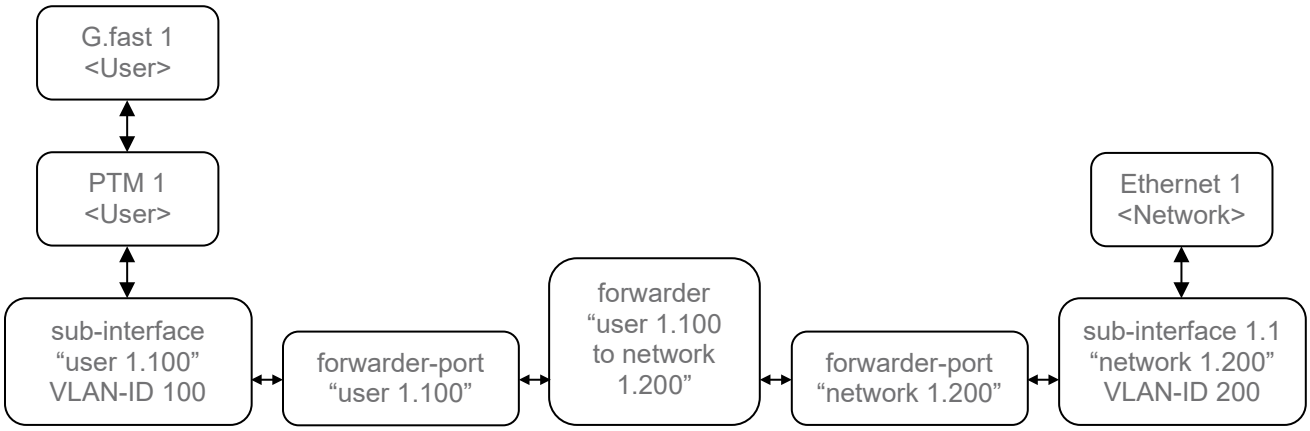

Figure 3 – Forwarder Ports

In the simplest use case of a 1:1 VLAN, this is all the forwarder needs to forward traffic between two sub-interfaces. The sub-interfaces determine which packets will be forwarded and how they will be manipulated. The forwarder just provides the means to associate the sub-interfaces.

7.2.2 Split Horizon Profiles

Once the interface usage is configured, a split horizon profile can be created and applied per forwarder to configure how traffic is forwarded between the various types of interfaces. Each profile specifies the usage of the ingress interface then lists the usages for which egress of the packet is not allowed from the ingress interface. For example, a profile could specify that for an ingress interface that is a user port, it is not allowed to send traffic to interfaces that are also user ports.

7.2.3 MAC Learning

In addition to the usage of an interface, the source and destination MAC addresses are key to making correct forwarding decisions for N:1 and N:M VLAN forwarding. Each forwarder contains configuration that determines how and if MAC source addresses are learned. It can also prevent traffic from being forwarded if it contains a certain MAC source address.

Once a MAC source address is learned, it is installed in the forwarding database for a given forwarder port. When a packet egresses a forwarder port, its MAC destination address is compared to the addresses in each of the other forwarder port's forwarding database to determine where the packet should be forwarded.

7.2.4 Flooding

In the case that the destination MAC address is not found in any forwarding database, it may be desired to flood the packet to all appropriate forwarder ports. To determine when and how this flooding occurs, a flooding policy profile can be created and associated with each forwarder. Each policy can be specified for a particular interface usage (e.g., user port) and/or a specific destination MAC address. It then assigns an appropriate action of either discarding the packet or flooding it to all interfaces of specified usage(s). For example, a forwarder may be configured to flood all packets with an unknown MAC address coming from a network port to all user ports.

7.3 Interface Usage

For the case of N:1 or N:M VLAN forwarding, the role each interface plays in the network is important to determine how traffic flow is managed. For example, in the context of an Access Node, traffic ingressing a user port should not normally be forwarded to another user port. Certain mechanisms as discussed above will be used to enforce this restriction. First, however, the way in which an interface is used must be explicitly known. For this the interface usage must be configured either by the user or by the system if the usage is already known. The 3 types of interface usage are:

- user port: The interface connects an Access Node to a user.
- network port: The interface connects an Access Node to a network.
- subtended-node port: The interface connects an Access Node to another Access Node.

Interface usage is applicable only to interface types derived from 'bbfift:ethernet-like' and for the following types and those derived from these types:

- 'ianaift:ethernetCsmacd'
- 'ianaift:ieee8023adLag'
- 'ianaift:ptm'
- 'bbfift:vlan-sub-interface'
- 'bbfift:l2-termination'.

7.4 Layer 2 Termination Interfaces

In some situations, Layer 2 VLAN traffic has to be terminated within a device and the device needs to manipulate the payload of the received frames, such as when IP packets are needed in the context of the in-band management NETCONF/YANG traffic, ANCP sessions, etc.

In the simple case, this can be realized at Layer 2 by creating a VLAN sub-interface on the network port and creating an IP interface on top of the VLAN sub-interface.

However, in other situations not all frames received via the VLAN sub-interface are to be processed by the device. For example, when the VLAN is a shared resource to manage a hub node and all its subtending nodes.

In this case the destination MAC address (and IP address) of some of the frames will have the address of the hub node while other frames will have the address of the subtending node. Consequently, received frames first need to be subject to a Layer 2 forwarding decision, whereas only frames addressed to the hub node are to be processed by the hub node at the IP layer.

Terminating Layer 2 after a forwarding decision is therefore modeled as a ‘Layer 2 termination interface’.

A Layer 2 termination interface differs from a VLAN sub-interface as it does not have a lower layer interface and hence also has no match criteria to classify frames received from this lower layer interface. It also has other characteristics that are used in the context of the common forwarding decisions, e.g., it has an interface usage that determines how forwarding decisions such as split-horizon-profile, flooding, etc. have to be taken inside the forwarder, i.e., it will have a QoS profile, ingress/egress VLAN tag manipulation, etc.

8 Ethernet-like Interfaces

There are several instances in the Common YANG modules where the interface list from ietf-interfaces [33] is augmented with a constraint on the types of interfaces to which the augmented nodes apply. For example, the type of an interface can be limited to the type of interfaces that transport Ethernet frames as shown below.

```
augment '/if:interfaces/if:interface' {
  when
    "derived-from-or-self(if:type, 'ianaift:ethernetCsmacd') or
     derived-from-or-self(if:type, 'ianaift:ieee8023adLag') or
     derived-from-or-self(if:type, 'ianaift:ptm') or
     derived-from-or-self(if:type, 'bbfift:vlan-sub-interface')" {
    description
      "Interfaces that can have QoS policy profiles assigned.";
  }
```

The augmentation to add a QoS policy reference to an interface is constrained to interfaces which are of one of four types or derived from those types. See RFC 7950 [35] for the full definition and usage of the derived-from-or-self() function.

Similarly, there are nodes which are references to an interface whose type is also constrained to those that transport Ethernet frames.

```
leaf interface {
  type if:interface-ref;
  must
    "derived-from-or-self(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'ianaift:ethernetCsmacd')
     or
     derived-from-or-self(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'ianaift:ieee8023adLag')
     or
     derived-from-or-self(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'ianaift:ptm')
     or
     derived-from-or-self(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'bbfift:sub-interface') ";
  mandatory true;
  description
    "References the lower-layer interface.";
}
```

In this example, the reference to the interface must be one of four types or derived from those types.

While this approach works well, it does not lend itself to extensibility when new interface types are defined either by the Broadband Forum, a vendor, an operator, or some other organization that is defining YANG data models. In order for these augments and must statements to be applicable to those interface types, either the new type or types need to be derived from one of these existing types or the new ones have to be added to the modeled constraints. This presents a challenge of keeping these models aligned and may not even be possible depending on the source of the newly defined interface type.

One solution that has been introduced is the creation of an abstract Ethernet type from which new interface types can be defined.

```
identity ethernet-like {
  base bbf-interface-type;
  description
    "An abstract identity defining a class of interfaces which
     represents a logical interface transporting Ethernet frames,
     i.e. frames with a destination and source MAC address, an
     Ethernet type or length field, and a payload. This
     'interface type' is intended only to be used to define
     constraints against a class of interfaces each of which have
     their 'type' derived from this identity (as well as potentially
     others). At no time should this identity be used as the 'type'
     for an interface.";
}
```

This abstract type is added to the constraints.

Updated augment example:

```
augment '/if:interfaces/if:interface' {
  when
    "derived-from-or-self(if:type, 'ianaift:ethernetCsmacd') or
     derived-from-or-self(if:type, 'ianaift:ieee8023adLag') or
     derived-from-or-self(if:type, 'ianaift:ptm') or
     derived-from-or-self(if:type, 'bbfift:vlan-sub-interface') or
     derived-from(if:type, 'bbfift:ethernet-like')” {
    description
      "Interfaces that can have QoS policy profiles assigned.";
  }
```

Updated must statement example:

```
leaf interface {
  type if:interface-ref;
  must
    "derived-from-or-self(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'ianaift:ethernetCsmacd')
     or
     derived-from-or-self(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'ianaift:ieee8023adLag')
     or
     derived-from-or-self(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'ianaift:ptm')
     or
     derived-from-or-self(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'bbfift:sub-interface')
     or
     derived-from(
     /if:interfaces/if:interface[if:name = current()]
     /if:type,'bbfift:ethernet-like') ";
  mandatory true;
  description
    "References the lower-layer interface.";
}
```

The use of ‘derived-from’ [35] stems from the identity’s definition which states it is an abstract identity which is not to be used as an actual interface type.

The method of using this abstract type is to add it as a base identity [35] to any Ethernet type definition that satisfies the definition of ‘ethernet-like’. For example,

```
identity new-ethernet-type {
  base bbfift:Ethernet-like;
  description
    "A new Ethernet type.";
}
```

By utilizing the abstract interface type, the Common YANG modules that define these constraints no longer have to be updated when a new Ethernet interface type is created.

9 Alarms

The intent of this section is to provide some general information regarding the alarm management.

Alarm management as applicable to access network equipment is based on the IETF “YANG Data Model for Alarm Management” (RFC 8632) [41], which defines a standardized alarm interface for network devices that can be easily integrated into management applications.

The design of this data model addresses usability requirements, such as those discussed in 3GPP TR 32.859 [50]; for example, improving the management of alarm overload through alarm shelving.

9.1 Alarms and Alarm Types

An alarm is an undesirable state of a resource.

In the “YANG Data Model for Alarm Management” [41] an instance of an alarm is thus uniquely identified by
- a fine-grained identification of the alarming resource, such as a specific interface
- an alarm type, which defines a possible undesirable state of the resource, such as ‘loss of signal’.

where alarm type is defined by
- an alarm type identifier (alarm-type-id)
- an alarm type qualifier (alarm-type-qualifier).

An alarm type identifier is modeled as a YANG identity, is defined at design time and can be abstract or concrete. Abstract alarms are a means of categorizing alarms and may also be used by a client for alarm filtering purposes.

An alarm type qualifier is a string that may be used, if the alarm type identifier alone cannot uniquely identify the alarm type, for example for alarms not known at design time.

As described in Section 3.2 of RFC 8632 [41], abstract alarms are generally not used for alarms. However, if an alarm is instrumented that was not known at design time, i.e., for which no concrete alarm type identifier has been defined in the YANG model, an abstract alarm type identifier qualified with an alarm type qualifier would be used. This practice, however, should be generally avoided to ensure that all possible alarms are known at design time.

9.1.1 Common Alarm Types

Alarm management for BBF Access Nodes defines a YANG identity hierarchy of common abstract alarm type identities to categorize alarm types defined by BBF applications based on the requirements of alarm reporting parameters associated with an alarm type as discussed in ITU-T X.733 [46], ITU-T X.736 [47], and 3GPP TS 32.111-2 [51]. These common abstract alarm type identities implicitly specify the alarm information (or alarm payload) defined in the IETF modules ietf-alarms and ietf-alarms-x733 that is applicable to alarm types based on these abstract alarm type identities.

The following abstract alarm type identifier identities are defined based on the identity alarm-type-id defined in ietf-alarms:

```
+--ietf-alarms:alarm-type-id
   +--bbf-alarm-types:bbf-alarm-type-id
      +--bbf-alarm-types:bbf-threshold-crossing-alarm-type-id
      +--bbf-alarm-types:bbf-security-alarm-type-id
```

The common abstract alarm types shown above do not define a fixed hierarchy of alarm types based on Event Types defined in ITU-T X.733 [46] and ITU-T X.736 [47], because the IETF "YANG Data Model for Alarm Management" (RFC 8632) [41] supports manageable Event Types for individual alarm types in ietf-alarms-x733. This allows a network operator to map the default vendor-specified Event Types associated with specific alarm types according to the operator's own requirements.

The alarm information to be associated with alarm type identities based on bbf-threshold-crossing-alarm-type-id and bbf-security-alarm-type-id is supported in the module ietf-alarms-x733.

Other abstract alarm type sub-categories may be defined to further categorize alarm types by BBF applications, but these sub-categories will be based directly or indirectly on one or more of the BBF alarm types listed above. Alarm types based on these sub-categories inherit the alarm information applicable to the alarm type on which they are based, but may refine this for abstract application-specific alarm types accordingly. Alarm type identities based on more than one of these abstract alarm type identities inherit the alarm information specification from each of these abstract identities.

Alarm information will be associated with each instance of an alarm and is implemented in ietf-alarms and ietf-alarms-x733 as nodes within the alarm list and shelved-alarm list and carried within the alarm-notification. Table 1 indicates whether this alarm information is mandatory (M), optional (O), or is not present (NP) for a concrete alarm type identifier when it is based on one or more of these abstract alarm type identifiers. If an abstract or concrete alarm type identifier is based on more than one abstract alarm type identifier, then the alarm information associated with that alarm type identifier will be a combination of the alarm information associated with each abstract alarm type identifier, where 'mandatory' has precedence over 'optional' has precedence over 'not present'.

Data node	Module	bbf-alarm-type-id	bbf-threshold-crossing-alarm-type-id	bbf-security-alarm-type-id
resource	ietf-alarms	M	M	M
alarm-type-id	ietf-alarms	M	M	M
alarm-type-qualifier	ietf-alarms	O	O	O
alt-resource	ietf-alarms	O	O	O
related-alarm	ietf-alarms	O	O	O
impacted-resource	ietf-alarms	O	O	O
root-cause-resource	ietf-alarms	O	O	O
time-created	ietf-alarms	M	M	M
is-cleared	ietf-alarms	M	M	M
last-raised	ietf-alarms	M	M	M
last-changed	ietf-alarms	M	M	M
perceived-severity	ietf-alarms	M	M	M
alarm-text	ietf-alarms	M	M	M
event-type	ietf-alarms-x733	O	O	O
probable-cause	ietf-alarms-x733	O	O	O
monitored-attributes	ietf-alarms-x733	O	O	O
proposed-repair-actions	ietf-alarms-x733	O	O	O
trend-indication	ietf-alarms-x733	O	O	O
backedup-status	ietf-alarms-x733	O	O	O

Data node	Module	bbf-alarm-type-id	bbf-threshold-crossing-alarm-type-id	bbf-security-alarm-type-id
backup-object	ietf-alarms-x733	O	O	O
additional-information	ietf-alarms-x733	O	O	O
threshold-information	ietf-alarms-x733	NP	M	NP
security-alarm-detector	ietf-alarms-x733	NP	NP	M
service-user	ietf-alarms-x733	NP	NP	M
service-provider	ietf-alarms-x733	NP	NP	M

Table 1 – Abstract BBF alarm types and associated alarm information

9.1.2 Application-specific Alarm Types

Concrete alarm types are to be defined by BBF applications and MUST either be based on at least one of the abstract alarm types defined in the module, bbf-alarm-types, or be based on an abstract alarm type defined by a BBF application and derived from at least one of those types.

As an example of an alarm hierarchy using abstract and concrete alarms is the following hierarchy of alarms defined in TR-355 [6]:

```
+--al:alarm-type-id                                              (abstract)
   +--bbf-alt:bbf-alarm-type-id                                  (abstract)
      +--bbf-fast-al:fast                                        (abstract)
         +--bbf-fast-al:fast-ftu-o-failures                      (abstract)
         |  +--bbf-fast-al:fast-ftu-o-line-initialization        (concrete)
         |  +--bbf-fast-al:fast-ftu-o-loss-of-signal             (concrete)
         |  +--bbf-fast-al:fast-ftu-o-loss-of-rmc                (concrete)
         |  +--bbf-fast-al:fast-ftu-o-loss-of-margin             (concrete)
         |  +--bbf-fast-al:fast-ftu-o-loss-of-power              (concrete)
         +--bbf-fast-al:fast-ftu-r-failures                      (abstract)
            +--bbf-fast-al:fast-ftu-r-loss-of-signal             (concrete)
            +--bbf-fast-al:fast-ftu-r-loss-of-rmc                (concrete)
            +--bbf-fast-al:fast-ftu-r-loss-of-margin             (concrete)
            +--bbf-fast-al:fast-ftu-r-loss-of-power              (concrete)
```

where al:alarm-type-id is the base alarm type identifier for all alarms managed by ietf-alarms and is located in the module ietf-alarms (RFC 8632) [41]; bbf-alt:bbf-alarm-type-id is the base alarm type identifier for all BBF defined alarms and is located in the module bbf-alarm-types. bbf-fast-al:fast is the base alarm type identifier for all FAST line alarms and is defined in module bbf-fast-alarms. This example also shows how application-specific alarms can be further categorized, i.e., FAST alarms are further categorized into alarms for local and remote FAST Transceiver Unit (FTU) failures.

10 Access Node Control Protocol

The intent of this section is to provide some general information regarding the use of the data model to manage the Access Node Control Protocol (ANCP) [2][30] on Access Nodes.

The ANCP YANG model published as part of this revision of the Technical Report supports only the topology discovery capability defined in RFC 6320 [30].

10.1 Partitions, Sessions and Adjacencies

A partition collects a set of access lines together that are to be managed together by one or more Network Access Server (NAS). In RFC 6320 [30] an Access Node (AN) may or may not support partitions. In the ANCP YANG model access lines on an AN that are to take part in ANCP must always be explicitly assigned to a partition. In the case where the AN does not support partitions, a single 'global' partition will need to be configured in the model.

If the single partition with partition-id = 'global' is configured, the PType and Partition ID in the ANCP Adjacency Message must be set to 0 (no partition) and 0 respectively.

A given partition may only collect access lines together of the same technology, e.g.,

- FastDSL

Configuration and operational state that depends on a specific technology, such as FastDSL, are augmented into the main ANCP YANG module bbf-ancp by technology-specific ANCP YANG modules.

An adjacency between the AN and a NAS for a given partition is managed in the model through the configuration of a session. The session manages the connection to the remote NAS, including which line attributes are to be reported in the Port Up and Port Down event messages sent by the session to the NAS.

The relationship between partitions, sessions and access lines, which are represented by interfaces, is shown below.

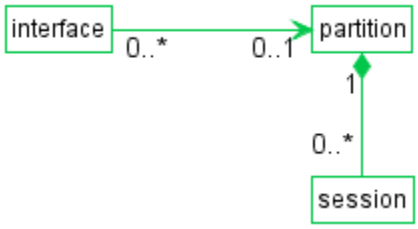

Figure 4 – Relationships between partitions, sessions and interfaces

The workflow to create an adjacency is as follows:

1. create a partition;
2. assign access lines to that partition;
3. create one or more sessions to manage an adjacency from the partition to a remote NAS.

10.1.1 Create a Partition

Before ANCP can be used on an AN at least one partition must be created. When creating the partition, the technology of the access lines to be assigned to the partition must be specified. An AN may support one or more technologies and this can be determined through the capabilities advertised by the AN.

10.1.2 Assigning Access Lines to a Partition

Access lines can be assigned to a partition through the interfaces in ietf-interfaces that represent the access lines. The assignment is made by referencing the partition from an interface. Configuring this reference automatically enables ANCP functionality for that access line.

10.1.3 Create a Session

For each adjacency between an AN and a NAS for a given partition, a session must be created for that partition.

The TCP connection to the NAS for which adjacency is to be attained is configured in the container 'network-access-server' in the list 'session'. This container also reports the identification of the remote NAS once adjacency has been achieved.

10.2 Topology Discovery

Topology discovery is enabled for the session by configuring at least one line attribute for the Port Up messages specific to the technology configured for the given partition. Line attributes are configured in the technology-specific containers found as child nodes to the container 'port-up' within the container 'topology-discovery' of a session. Similarly, line attributes to be included in Port Down messages are configured in the technology-specific containers found as child nodes to the container 'port-down' within the container 'topology-discovery' of a session.

10.3 Access Line Identification

The identification of an access lines on the AN must be configured and is defined through a combination of logical port information on the user side as well as on the NAS side of the AN. RFC 6320 [30] defines four ANCP TLVs for access line identification:

- Access-Loop-Circuit-ID
- Access-Loop-Remote-ID
- Access-Aggregation-Circuit-ID-Binary
- Access-Aggregation-Circuit-ID-ASCII.

Access line identification is also required to be supported for DHCP and PPPoE as specified in TR-101 Issue 2 Section 3.9 [1]. To ensure a consistent identification of access lines across ANCP, DHCP and PPPoE, common access line identification parameters can be configured within a subscriber profile, which is assigned to the VLAN sub-interface associated with the access line.

The specific access line identification TLVs to be sent in ANCP messages for a given session is configured in the leaf-list 'line-identification' within the container 'access-line-identification'.

10.3.1 Access-Loop-Circuit-ID

The value inserted into this TLV must be the value configured for the leaf 'circuit-id' within the subscriber-profile that is referenced from the VLAN sub-interface associated with the access line. If no such subscriber-profile

has been configured, then a TLV must be generated according to the syntax defined in the leaf ‘access-loop-circuit-id’ within the container ‘access-line-identification’ of the session. If this leaf is also not defined, then an empty TLV must be inserted, i.e., a TLV with Length = 0.

10.3.2 Access-Loop-Remote-ID

The value inserted into this TLV must be the value configured for the leaf ‘remote-id’ within the subscriber-profile that is referenced from the VLAN sub-interface associated with the access line. If no such subscriber-profile has been configured, then an empty TLV must be inserted, i.e., a TLV with Length = 0.

10.3.3 Access-Aggregation-Circuit-ID-Binary and Access-Aggregation-Circuit-ID-ASCII

Access-Aggregation-Circuit-ID-Binary identifies or partially identifies a specific access line by means of the VLAN IDs of the inner and outer VLAN tags of the data frames coming from that access line on the NAS side of the AN. The format of Access-Aggregation-Circuit-ID-Binary is specified in RFC 6320 [30].

Access-Aggregation-Circuit-ID-ASCII is an ASCII equivalent of Access-Aggregation-Circuit-ID-Binary TLV, the format of which is explicitly configured in ‘access-aggregation-circuit-id-ascii’ within the container ‘access-line-identification’ of a session. As per RFC 6320 [30], it shall contain VLAN IDs, e.g., 'S-VID:C-VID’, but may contain any characters and variables in a format as specified by TR-101 Issue 2 Section 3.9.3 [1].

If frames received on the subscriber interface are forwarded to multiple VLAN sub-interfaces, then the AN would need to know how to select which VLAN sub-interface to use to derive the VLAN-IDs for Access-Aggregation-Circuit-ID-Binary and Access-Aggregation-Circuit-ID-ASCII. This information is configured in through the choice ‘access-aggregation-circuit-id’ within the container ‘ancp’ on the interface representing the access line. The choice has two cases

‘auto-derived’	the VLAN IDs are determined from the VLAN sub-interface that classifies ingress frames with the lowest VLAN ID value, combined with the related forwarding and network-side VLAN sub-interface configuration.
‘explicit’	up to two VLAN IDs can be explicitly configured.

10.3.4 Additional Formatting

If the Access-Loop-Circuit-ID or Access-Aggregation-Circuit-ID-ASCII use the variables for a slot, a port or other numbered variable, the configuration ‘start-numbering-from-zero’ controls whether the number begins with 0 or 1 and ‘use-leading-zero’ whether or not leading zeroes are to be used when representing the numbers.

10.3.5 Supporting FastDSL Bonding

For a bonding group that bonds multiple access lines, a primary line for the bonding group must be selected, which will be used to generate the Access-Loop-Circuit-ID. This is configured in the leaf ‘primary-line’ of a bonding group interface defined in module bbf-gbond, first available in TR-355 Amendment 3 [6].

10.4 Controlling Port Messages

The ANCP model also supports additional configurations to control how and when Port Up and Port Down event messages are sent.

10.4.1 Threshold-based Reporting

For some specific line attributes, TR-301 [5] requires that if the measurement on the port changes by more than a configurable threshold, the port state must be reported to the PMA. In the ANCP model shift-up and shift-down thresholds can be configured for specific line attributes per partition, applying to all sessions of that partition.

The configuration is made in the technology-specific containers ‘vdsl’ and ‘fast’ containers within the container ‘port-message-control’ of a partition.

10.4.2 Delaying the Initial Port Up Message

Unstable connections that go in and out of sync and line characteristics that are unstable during the synchronization process can cause a flood of Port Up and Port Down messages.

To limit unnecessary Port Up and Port Down event messages during the synchronization process, it is possible to configure an ‘initial-port-up-delay’ which requires that the line be synchronized for a given period, before the first Port Up message is sent following synchronization of the line.

The configuration is made in the technology-specific containers ‘vdsl’ and ‘fast’ within the container ‘port-message-control’ of a partition.

10.4.3 Dampening Mechanism

Seamless Rate Adaptation (SRA) and Fast Rate Adaptation (FRA) of FastDSL access lines may result in rapid and continuous data rates changes. RFC 6320 [30] recommends that a dampening mechanism be supported to limit the rate at which state changes of access lines are reported to the NAS. This is supported in the ANCP YANG model through the configuration of a ‘port-up-port-down-withholding-interval’ within the container ‘port-message-control’ of a partition.

The withholding interval applies to each access line independently and defines an interval which begins when a Port Up message is sent for that access line. During this interval no further Port Up message will be sent to the NAS for that given access line. If, at the end of the withholding interval, there has been a change in line state of the given line to that when the last Port Up message was sent, a Port Up message is sent with the new state (with the withholding interval for that line applying again).

10.5 Statistics

The model supports the reporting of statistics per session for adjacency messages sent and received and statistics for topology discovery messages sent per access line per session.

10.6 Alarms and the Operational State of a Session

If a session encounters an issue and is unable to operate correctly as intended by the configuration, the session will raise an alarm to the Network Management System (NMS). The YANG data model supports the following alarms:

- ‘adjacency-failure’
- ‘capability-negotiation-failure’
- ‘message-type-not-supported’
- ‘nas-not-reachable’.

If an alarm is raised, the session may cease operation and come to a halt, and as a result may need to be reset by the NMS once the issue has been resolved.

Figure 5 below summarizes the behavior of an ANCP session from the viewpoint of the operation state defined in the YANG data model, illustrating how alarms impact states and how or when these alarms may be cleared.

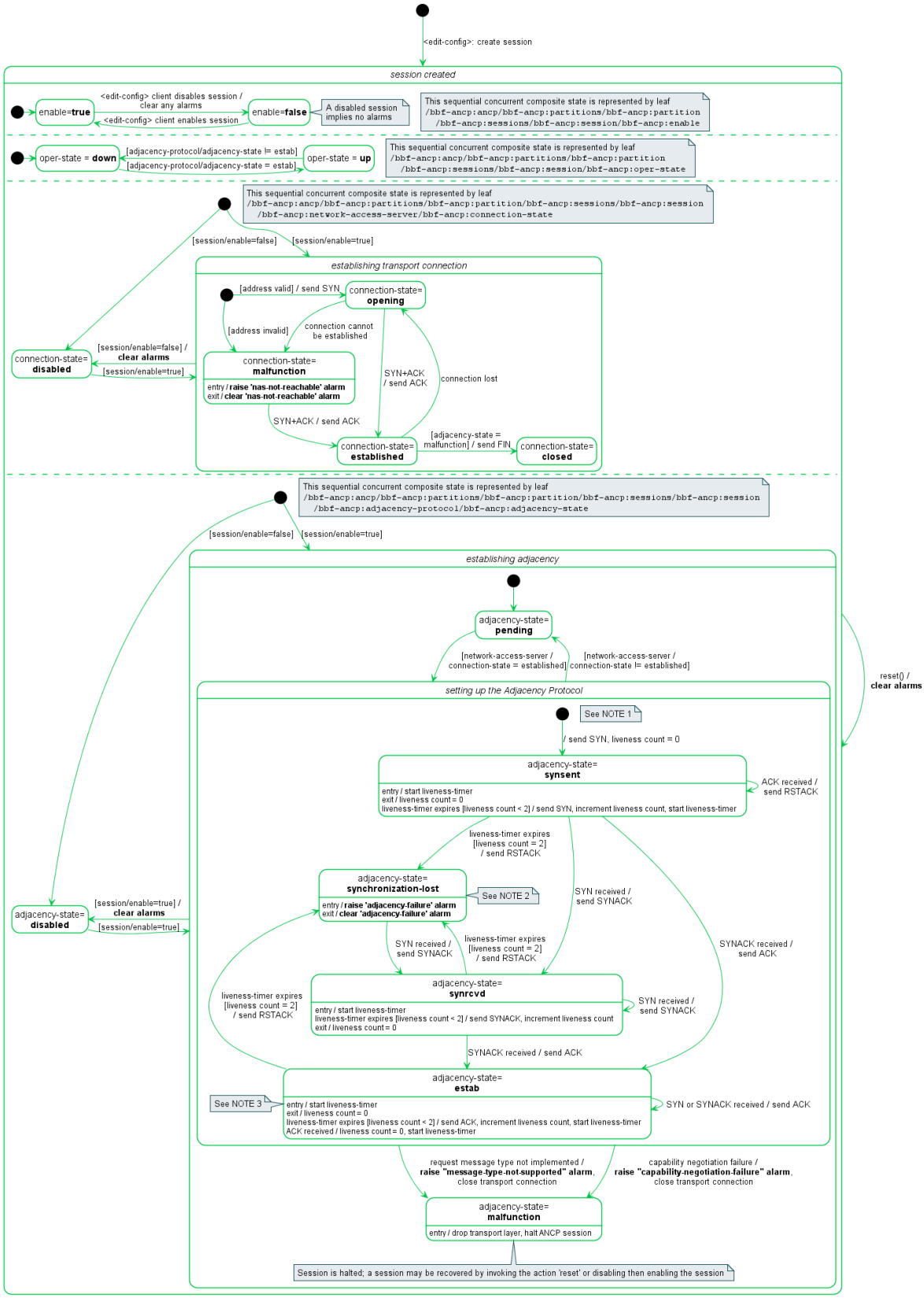

Figure 5 – Statechart of an ANCP session (informational only)

NOTE 1: Only significant state transitions are shown that are key to understanding the behavior of the YANG data model.

NOTE 2: This state is entered when a session declares a loss of synchronization as defined in section 3.5.2.7 of RFC 6320 [29]. RFC 6320 [30] is not specific as to how a session re-establishes synchronization; how this is achieved does not need specific representation in the YANG data model.

NOTE 3: Section 3.5.2.2 of RFC 6320 [30] specifies that no more than one ACK should be sent within any time period of length defined by the liveness timer.

11 Software Management

The intent of this section is to provide some general information regarding the use of the Software Management model. Although the main requirements driving the model were to enable the PMA to manage software images on a DPU according to requirement R-214 of TR-301 [5] and to support the Software Image Management requirements of ITU-T G.997.2 Annex S [44] and ITU-T G.988 clause 9.1.4 and Appendix I [42], the model has nevertheless been intentionally designed to generally support the management of software on any hardware component supported by a device. It is, therefore, not possible here to describe every possible application of the model, but rather the information below provides the theory behind the model and illustrates some general use cases.

The management of software as applicable to access network equipment is based on the IETF “A YANG Data Model for Hardware Management” (RFC 8348) [37].

11.1 Components, Software and Revisions

The Software Management model is based on the concept that a hardware component of a device may support one or more pieces of manageable software that may be replaced during the lifetime of the device, whereby each software is defined by a set of revisions of that software that is present on the component. Since the model does not make any assumptions about what the ‘software’ is, the ‘software’ could be an executable code image or other operating information, such as a script or even, for example, a text-based profile. The format of software files and how they are processed, such as extracting software image from an archive, is assumed to be an integral part of the download activity and as such is an implementation aspect not within the scope of this Technical Report.

Consequently, the Software Management model augments a ‘component’ within the module ietf-hardware [37] with a container ‘software’ containing a list ‘software’. Each entry in the list is uniquely identified by a name (string), whereby this name should also be representative of its purpose and identify its association with the hardware component, e.g., “firmware”. The maximum number of revisions supported for a specific software would be device- and software-specific.

Each revision is defined by the following metadata:

id	A unique identifier for the software revision.
alias	If supported by the specific software, an optional unique name for the software revision that may be assigned by the client when the software revision is downloaded.
version	A string that identifies the published version of the revision.
is-committed	If supported by the specific software, a Boolean that if ‘true’ indicates that the revision will be loaded and made active upon a reboot of the device.
is-active	If supported by the specific software, a Boolean that if ‘true’ indicates that the revision is currently loaded and active.
is-valid	If supported by the specific software, a Boolean that if ‘true’ indicates that the revision’s contents have been verified to be a valid revision that may be activated and committed.
product-code	An optional string to indicate vendor-specific product code information.
hash	An optional hash of the revision

At any given time, at most one revision may be active and at most one revision may be committed.

Figure 6 illustrates the relationship between hardware components and software within the YANG data model.

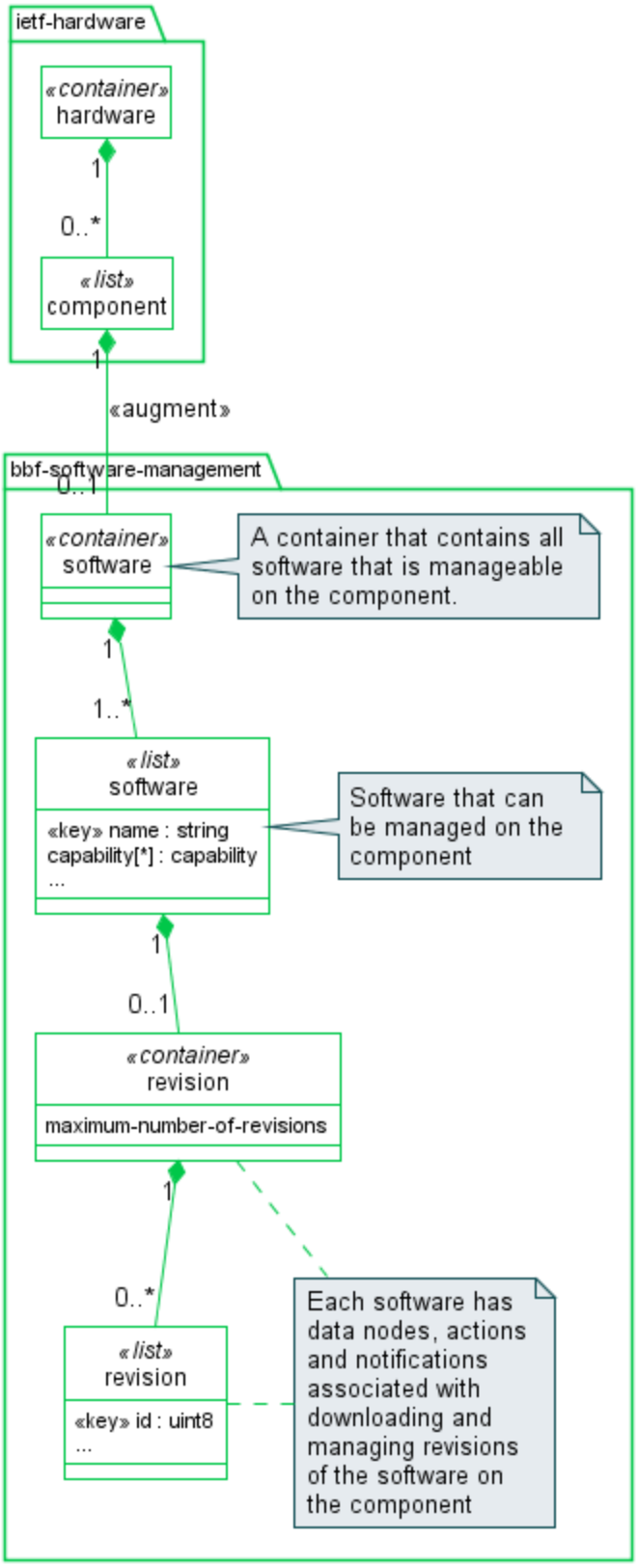

Figure 6 – UML diagram showing the relationship between software and hardware components

11.2 Management Capabilities

As discussed above, the Software Management data model does not make any assumptions about the software. Consequently, the management capabilities for a software as implemented by a device may vary dependent on the specific type of software or specific implementation. For example, in some cases revisions may be implemented as fixed entities in the device, which may neither be created nor deleted, but only replaced, such as Network Termination (NT) software images managed according to ITU-T G.997.2 Annex S [44]. For some types of software, activating and committing the software may not apply. To enable clients to determine the management capabilities supported by a specific software, the data model enables an implementation to advertise the management capabilities in the leaf 'capability' for each software.

A device may advertise the following capabilities.

activate The software supports activating software revisions. Whether a revision is currently activated is reported in the leaf 'is-active'. See Section 11.3.2 below.

alias The software supports the association of a unique alias to software revisions.

commit The software supports committing software revisions. Whether a revision is currently committed is reported in the leaf 'is-committed'. See Section 11.3.3 below.

conditional-activation
The software supports the conditional activation of software revisions applicable to voice calls; this requires the support of the module bbf-software-management-voice. See Section 11.4 below.

delete The software supports deleting software revisions. See Section 11.3.4 below.

download-target-selection-by-system
The software supports automatic selection of the revision to be used as the target of a download by the system, see Section 11.3.1 below.

download-target-selection-by-user
The software supports the client selecting a specific revision to be used as the target of a download, see Section 11.3.1 below.

validate The device supports an explicit validation of the software during download and reports this in the leaf 'is-valid'.

NOTE: To support downloading of revisions, at least one of the capabilities 'download-target-selection-by-system' or 'download-target-selection-by-user' must be supported.

Depending on the specific capabilities advertised by a software, specific actions and notifications may or may not be available for that software. See Section 11.3 for more details.

11.3 Managing Revisions

Depending on the management capabilities of a specific software, a client manages revisions of that software by downloading, activating, and committing the software, which moves the revisions through a series of states as shown in Figure 7.

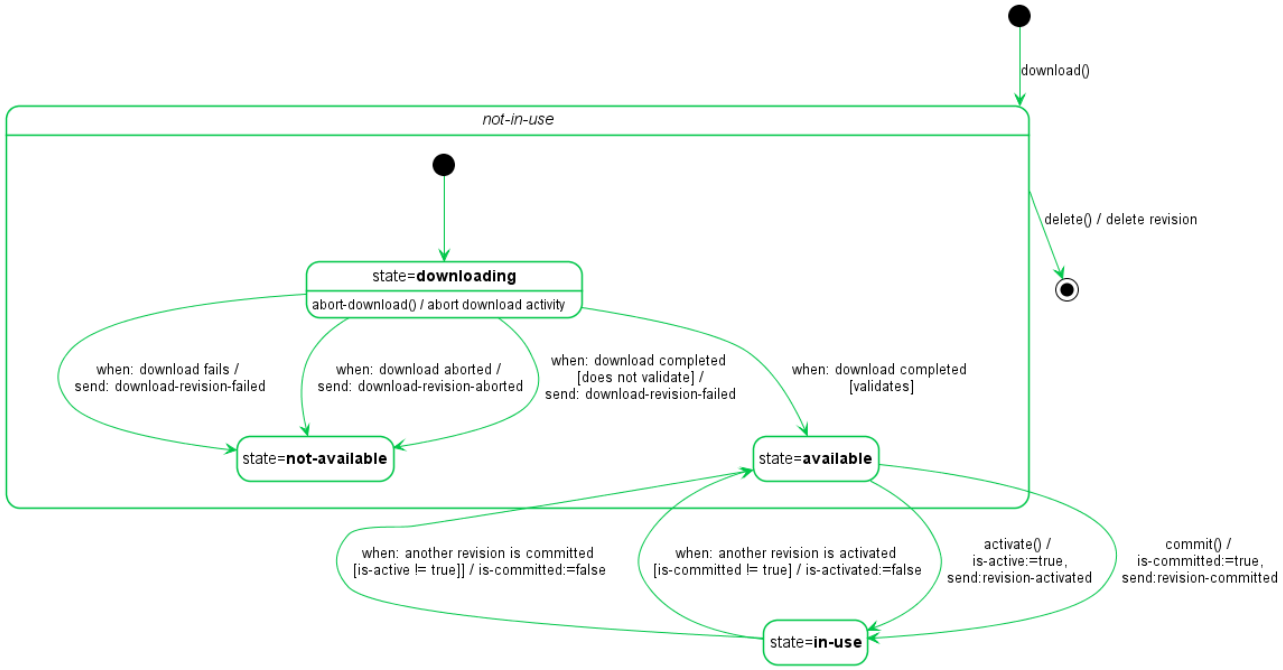

Figure 7 – State machine of a revision

11.3.1 Download

If a software supports downloading revisions, it must advertise at least one of the capabilities 'download-target-selection-by-system' or 'download-target-selection-by-user'. If a software does not support download, it may support a fixed set of revisions that may be managed, but this is not within the scope of this Technical Report.

The download of revisions is managed within the container 'download' within the container 'revisions', which is a child node of the list 'software'. Only a single download activity per software is supported at any one time: the container 'download' will not be available in the operational state datastore if a download is in progress.

To initiate an activity to download a revision of a software to a component, the client invokes the action 'download'.

The choice 'source' supports a single case: 'url'. However, vendors may augment this choice to support additional download methods, but this is beyond the scope of this Technical Report.

The choice 'target' gives the client the opportunity to specify how an entry in the list 'revision' is to be selected to store the downloaded revision. Depending on the management capabilities of the specific software, the client may select a specific target revision for the download or let the system select the target revision. Depending on the selection, a new entry in the list 'revision' may be created or an existing entry replaced.

Important to the concept of the Software Management data model is that a successful invocation of the action 'download' must always result in the device creating or replacing an entry in the list 'revision' with the value of the leaf 'id' of that revision as specified in the output element of the RPC, regardless of whether the activity triggered by the action is successful or not. If, for example, the revision cannot be downloaded, is corrupt or not compatible with the software, the revision must enter the state 'not-available'.

An entry must not be created or replaced in the list 'revision' if the invocation of the action is unsuccessful.

If an implementation of the model limits the maximum number of entries in the list ‘revision’, the known maximum number of supported entries should be reported in the leaf ‘maximum-number-or-revisions’ within the container ‘revisions’.

A client can abort an on-going download by invoking the action ‘abort-download’ within the container ‘abort-download’ of an entry in the list ‘revision’.

The client is notified of the results of the activities initiated by the actions discussed above through the notifications ‘revision-downloaded’, ‘download-revision-failed’, and ‘download-revision-aborted’.

The leaf ‘state’ of a revision indicates whether the download activity is ongoing or, if it has completed, the outcome of the download as shown in Figure 7 above.

11.3.2 Activate

When this action is applied to a software revision, execution of the software revision that is currently active will be suspended and the device will load and execute the software represented by the given revision, whereby any steps necessary to achieve this are performed implicitly by the device; such steps are not within the scope of this Technical Report.

If a software supports explicitly activating revisions, it must advertise the capability ‘activate’. If a software does not support ‘activate’, loading/execution of the software would be implementation-specific and is not within the scope of this Technical Report.

The activation of revisions is managed within the container ‘activate’ of an entry in the list ‘revision’ and is only available in the operational state datastore for software that supports the capability ‘activate’ and for revisions of that software that are in the state ‘available’ or ‘in-use’.

To initiate an activity to activate a revision of a software on a component, the client invokes the action ‘activate’.

If a restart of the component or device is necessary to activate a specific revision of a software, this must be performed implicitly by the device as part of the activation activity.

The client is notified of the results of the activities initiated by the action ‘activate’ through the notifications ‘revision-activated’ and ‘activate-failed’.

11.3.3 Commit

When this action is applied to a software revision, the software represented by this revision will be loaded and executed by the device upon subsequent restarts of the device or component.

If a software supports explicitly committing revisions, it must advertise the capability ‘commit’. If a software does not support ‘commit’, whether a specific software revision is loaded and executed at the startup of the device or component would be implementation-specific and is not within the scope of this Technical Report.

Committing revisions is managed within the container ‘commit’ of an entry in the list ‘revision’ and is only available in the operational state datastore for software that supports the capability ‘commit’ and for revisions of that software that are in the state ‘available’ or ‘in-use’.

To initiate an activity to commit a revision of a software on a component, the client invokes the action ‘commit’.

The client is notified of the results of the activities initiated by the action ‘commit’ through the notifications ‘revision-committed’ and ‘commit-failed’.

11.3.4 Delete

When this action is applied to a software revision, the revision will be deleted and no longer available on the component.

If a software supports explicitly deleting revisions, it must advertise the capability ‘delete’. If a software supports downloading revisions but not deleting revisions, then revisions would only be able to be replaced when downloading a revision.

Deleting revisions is managed within the container ‘delete’ of an entry in the list ‘revision’ and is only available in the operational state datastore for software that supports the capability ‘delete’ and for revisions of that software that are not in the state ‘in-use’.

To initiate an activity to delete a revision of a software on a component, the client invokes the action ‘delete’.

The client is notified of the completion of the deletion through the notifications ‘revision-deleted’ and ‘delete-failed’.

11.4 Supporting the Management of Software Upgrade Processes of FastDSL NTs

To support the management of the software upgrade processes in FastDSL Network Terminations (NT) as defined in ITU-T G.997.2. Annex S [44], a device populates a ‘remote’ hardware component within the operational state datastore of IETF Hardware Management data model for each NT that it detects as described below:

When an NT is detected on an interface, the device performs the following in the operational state datastore:

- instantiates a component of class ‘bbf-hwt:fastdsl-nt’;
- on the interface on which the NT was detected, instantiates a reference to this component define in the module bbf-interfaces-remote-hardware;
- if the NT supports the management of its software images as supported by the device,
    - instantiates an entry in the software list of the component for each software on the remote device that can be managed by the device;
    - for each software,
        - populates the leaf-list ‘capability’ with the software management capabilities supported for that software; see Table 2 below for the set of capabilities required to support the management of NT software images according to ITU-T G.997.2 Annex S [44];
        - instantiates an entry in the revision list for each revision that it detects on the NT;
- populates the leafs ‘mfg-name’, ‘model-name’, ‘serial-num’ of the fastdsl-nt component from the inventory data provided by the NT;
- optionally populates other optional nodes of the component as supported by the vendor.

When an NT disconnects, the device

- deletes the component representing the NT in the IETF Hardware Management data model.

Capability	Remarks
activate	Mandatory as this is a requirement of the software upgrade processes defined in the ITU-T recommendations.
alias	Not required as the software images defined in the ITU.T recommendations do not support an alias.
commit	Mandatory as this is a requirement of the software upgrade processes defined in the ITU-T recommendations.
conditional-activation	Not applicable to the software upgrade process defined in ITU-T G.988. Optional for the software upgrade process defined in ITU-T G.997.2 Annex S. See NOTE 1.
delete	Cannot be supported, because 2 software images are always automatically instantiated as a requirement of the ITU-T recommendations.
download-target-selection-by-system	Recommended to relieve the client of the need to track which 'id' is available for a download and leave this to the server. See NOTE 2.
download-target-selection-by-user	Optional. See NOTE 2.
validate	Mandatory as this is a requirement of the software upgrade processes defined in the ITU-T recommendations.

Table 2 – Capabilities required to support the management of FastDSL NT software images according to ITU-T G.997.2 Annex S

NOTE 1: To support the requirement of G.997.2 Annex S clause S.7.1.2.1 [44], the module bbf-software-management-voice will need to be supported by the server. This module augments additional data nodes into the input element of the 'activate' action of a revision to enable the client to specify the conditions under which the activation request for a software image shall be executed dependent on whether voice calls are currently in progress.

NOTE 2: At least one of the capabilities 'download-target-selection-by-system' and 'download-target-selection-by-user' must be supported.

End of Broadband Forum Technical Report TR-383