A YANG Data Model for Microwave Radio Link

draft-mwdt-ccamp-mw-yang-01

Abstract

This document defines a YANG data model in order to control and manage the radio link interfaces, and the connectivity to packet (typically Ethernet) interfaces in a microwave/millimeter wave node.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 23, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Terminology and Definitions

The following terms are used in this document:

Carrier Termination (CT) is an interface for the capacity provided over the air by a single carrier. It is typically defined by its transmitting and receiving frequencies.

Radio Link Terminal (RLT) is an interface providing packet capacity and/or TDM capacity to the associated Ethernet and/or TDM interfaces in a node and used for setting up a transport service over a microwave/millimeter wave link.

The following acronyms are used in this document:
ACM  Adaptive Coding Modulation
ATPC Automatic Transmit Power Control
CM   Coding Modulation
CT   Carrier Termination
RLT  Radio Link Terminal
RTPC Remote Transmit Power Control
XPIC Cross Polarization Interference Cancellation
2. Introduction

This document defines a YANG data model for management and control of the radio link interface(s) and the relationship to packet (typically Ethernet) and/or TDM interfaces in a microwave/millimeter wave node. The data model includes configuration and state data.

The design of the data model follows the framework for management and control of microwave and millimeter wave interface parameters defined in [mw-fmwk]. This framework identifies the need and the scope of the YANG data model, the use cases and requirements that the model needs to support. Moreover, it provides a detailed gap analysis to identify the missing parameters and functionalities of the existing and established models to support the specified use cases and requirements, and based on that recommends how the gaps should be filled with the development of the new model.

According to the conclusion of the gap analysis, the structure of the data model is based on the structure defined in [I-D.ahlberg-ccamp-microwave-radio-link] and it augments RFC 7223 to align with the same structure for management of the packet interfaces. More specifically, the model will include interface layering to manage the capacity provided by a radio link terminal for the associated Ethernet and TDM interfaces, using the principles for interface layering described in RFC 7223 as a basis.

The designed YANG data model uses the IETF: Radio Link Model [I-D.ahlberg-ccamp-microwave-radio-link] and the ONF: Microwave Modeling [ONF-model] as the basis for the definition of the detailed leafs/parameters, and proposes new ones to cover identified gaps which are analyzed in [mw-fmwk].

3. YANG Data Model (Tree Structure)
module: ietf-microwave-radio-link

+--rw radio-link-protection-groups
    +--rw radio-link-protection-group* [name]
        +--rw name                                      string
        +--rw protection-architecture-type?              identityref
        +--rw protection-operation-type?                 enumeration
        +--rw working-entity*                            if:interface-ref
        +--rw revertive-wait-to-restore?                 uint16
        +--rw radio-link-protection-members*             if:interface-ref
        +++-x protection-external-commands
            +---w input
                +---w protection-external-command?           identityref

+--ro radio-link-protection-groups-state
    +--ro radio-link-protection-group* [name]
        +--ro name                                      string
        +--ro protection-status?                        identityref

+--rw xpic-pairs {xpic}?
    +--rw xpic-pair* [name]
        +--rw name                                      string
        +--rw enabled?                                  boolean
        +--rw xpic-members*                             if:interface-ref

+--rw mimo-groups {mimo}?
    +--rw mimo-group* [name]
        +--rw name                                      string
        +--rw enabled?                                  boolean
        +--rw mimo-members*                             if:interface-ref

augment /if:interfaces/if:interface:
    +--rw id?                                        string
    +--rw mode                                       identityref
    +--rw carrier-terminations*                      if:interface-ref
    +--rw rlp-groups*                                -> /radio-link-protection-groups
                                                   /radio-link-protection-group/name
    +--rw xpic-pairs*                                -> /xpic-pairs/xpic-pair/name {xpic}?
    +--rw mimo-group?                                -> /mimo-groups/mimo-group/name {mimo}?
    +--rw tdm-connections* [tdm-type] {tdm}?
        +--rw tdm-type                                  identityref
        +--rw tdm-connections                            unit16

augment /if:interfaces/if:interface:
    +--rw carrier-id?                                string
    +--rw tx-enabled?                                boolean
    +--rw tx-frequency                               uint32
    +--rw rx-frequency?                              uint32
Internet-Draft    Microwave YANG Model        December 2016

++--rw rx-frequency-config?        boolean
++--rw duplex-distance              uint32
++--rw channel-separation           decimal64
++--rw polarization?                enumeration
++--rw power-mode                   enumeration
++--rw selected-output-power        power
++--rw atpc-lower-threshold         power
++--rw atpc-upper-threshold         power
++--rw coding-modulation-mode       enumeration
++--rw selected-cm                  identityref
++--rw selected-min-acm             identityref
++--rw selected-max-acm             identityref
++--rw if-loop?                     enumeration
++--rw rf-loop?                     enumeration
++--rw ct-performance-thresholds
  ++--rw received-level-alarm-threshold?  power
  ++--rw transmitted-level-alarm-threshold?  power
  ++--rw ber-alarm-threshold?             enumeration

augment /if:interfaces-state/if:interface:
  ++--ro tx-oper-status?              enumeration
  ++--ro actual-transmitted-level?     power
  ++--ro actual-received-level?        power
  ++--ro actual-tx-cm?                 identityref
  ++--ro actual-snir?                  decimal64
  ++--ro actual-xpi?                   decimal64  [xpic]?
  ++--ro capabilities
    ++--ro min-tx-frequency?            uint32
    ++--ro max-tx-frequency?            uint32
    ++--ro min-rx-frequency?            uint32
    ++--ro max-rx-frequency?            uint32
    ++--ro available-min-output-power?  power
    ++--ro available-max-output-power?  power
    ++--ro available-min-acm?           identityref
    ++--ro available-max-acm?           identityref

augment /if:interfaces-state/if:interface/if:statistics:
  ++--ro bbe?                        yang:counter32
  ++--ro es?                         yang:counter32
  ++--ro ses?                        yang:counter32
  ++--ro uas?                        yang:counter32
  ++--ro min-rltm?                   power
  ++--ro max-rltm?                   power
  ++--ro min-trlm?                   power
  ++--ro max-trlm?                   power

4. YANG Module

```yang
<CODE BEGINS> file "ietf-microwave-radio-link.yang"

module ietf-microwave-radio-link {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-microwave-radio-link";
  prefix mrl;

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-interfaces {
    prefix if;
  }

  import iana-if-type {
    prefix ianaift;
  }

  organization
    "IETF CCAMP Working Group";
  contact
    "jonas.ahlberg@ericsson.com
     amy.yemin@huawei.com
     Xi.Li@neclab.eu
     cjbc@it.uc3m.es
     k-kawada@ah.jp.nec.com";

  description
    "This is a module for the entities in a generic microwave system.";

  revision 2016-12-22 {
    description
      "Draft revision covering a complete scope for configuration and state data for radio link interfaces.";

    reference "";
  }
```

revision 2016-10-29 {
    description
    "Draft revision.";
    reference "";
}

/*
 * Features
 */

feature xpic {
    description
    "Indicates that the device supports XPIC.";
}

feature mimo {
    description
    "Indicates that the device supports MIMO.";
}

feature tdm {
    description
    "Indicates that the device supports TDM.";
}

/*
 * Interface identities
 */

identity radio-link-terminal {
    base ianaift:iana-interface-type;
    description
    "Interface identity for a radio link terminal.";
}

identity carrier-termination {
    base ianaift:iana-interface-type;
    description
    "Interface identity for a carrier termination.";
}

/*
 * Radio-link-terminal mode identities
 */
identity rlt-mode {
  description
    "A description of the mode in which the radio
     link terminal is configured. The format is X
     plus Y. X represent the number of bonded
     carrier terminations. Y represent the number
     of protecting carrier terminations.";
}

identity one-plus-zero {
  base rlt-mode;
  description
    "1 carrier termination only.";
}

identity one-plus-one {
  base rlt-mode;
  description
    "1 carrier termination
     and 1 protecting carrier termination.";
}

identity two-plus-zero {
  base rlt-mode;
  description
    "2 bonded carrier terminations.";
}

/*
 * Coding and modulation identities
 */

identity coding-modulation {
  description
    "The coding and modulation schemes.";
}

identity half-bpsk-strong {
  base coding-modulation;
  description
    "Half BPSK strong coding and modulation scheme.";
}
identity half-bpsk {
    base coding-modulation;
    description
        "Half BPSK coding and modulation scheme.";
}

identity half-bpsk-light {
    base coding-modulation;
    description
        "Half BPSK light coding and modulation scheme.";
}

identity bpsk-strong {
    base coding-modulation;
    description
        "BPSK strong coding and modulation scheme.";
}

identity bpsk {
    base coding-modulation;
    description
        "BPSK coding and modulation scheme.";
}

identity bpsk-light {
    base coding-modulation;
    description
        "BPSK light coding and modulation scheme.";
}

identity qpsk {
    base coding-modulation;
    description
        "QPSK coding and modulation scheme.";
}

identity qam-4-strong {
    base coding-modulation;
    description
        "4 QAM strong coding and modulation scheme.";
}

identity qam-4 {
    base coding-modulation;
    description
        "4 QAM coding and modulation scheme.";
}
identity qam-4-light {
  base coding-modulation;
  description
    "4 QAM light coding and modulation scheme.";
}

identity qam-16-strong {
  base coding-modulation;
  description
    "16 QAM strong coding and modulation scheme.";
}

identity qam-16 {
  base coding-modulation;
  description
    "16 QAM coding and modulation scheme.";
}

identity qam-16-light {
  base coding-modulation;
  description
    "16 QAM light coding and modulation scheme.";
}

identity qam-32-strong {
  base coding-modulation;
  description
    "32 QAM strong coding and modulation scheme.";
}

identity qam-32 {
  base coding-modulation;
  description
    "32 QAM coding and modulation scheme.";
}

identity qam-32-light {
  base coding-modulation;
  description
    "32 QAM light coding and modulation scheme.";
}

identity qam-64-strong {
  base coding-modulation;
  description
    "64 QAM strong coding and modulation scheme.";
}
identity qam-64 {
    base coding-modulation;
    description
        "64 QAM coding and modulation scheme.";
}

identity qam-64-light {
    base coding-modulation;
    description
        "64 QAM light coding and modulation scheme.";
}

identity qam-128-strong {
    base coding-modulation;
    description
        "128 QAM strong coding and modulation scheme.";
}

identity qam-128 {
    base coding-modulation;
    description
        "128 QAM coding and modulation scheme.";
}

identity qam-128-light {
    base coding-modulation;
    description
        "128 QAM light coding and modulation scheme.";
}

identity qam-256-strong {
    base coding-modulation;
    description
        "256 QAM strong coding and modulation scheme.";
}

identity qam-256 {
    base coding-modulation;
    description
        "256 QAM coding and modulation scheme.";
}

identity qam-256-light {
    base coding-modulation;
    description
        "256 QAM light coding and modulation scheme.";
}

identity qam-512-strong {
    base coding-modulation;
    description
    "512 QAM strong coding and modulation scheme.";
}

identity qam-512 {
    base coding-modulation;
    description
    "512 QAM coding and modulation scheme.";
}

identity qam-512-light {
    base coding-modulation;
    description
    "512 QAM light coding and modulation scheme.";
}

identity qam-1024-strong {
    base coding-modulation;
    description
    "1024 QAM strong coding and modulation scheme.";
}

identity qam-1024 {
    base coding-modulation;
    description
    "1024 QAM coding and modulation scheme.";
}

identity qam-1024-light {
    base coding-modulation;
    description
    "1024 QAM light coding and modulation scheme.";
}

identity qam-2048-strong {
    base coding-modulation;
    description
    "2048 QAM strong coding and modulation scheme.";
}

identity qam-2048 {
    base coding-modulation;
    description
    "2048 QAM coding and modulation scheme.";
}
identity qam-2048-light {
  base coding-modulation;
  description
    "2048 QAM light coding and modulation scheme.";
}

identity qam-4096-strong {
  base coding-modulation;
  description
    "4096 QAM strong coding and modulation scheme.";
}

identity qam-4096 {
  base coding-modulation;
  description
    "4096 QAM coding and modulation scheme.";
}

identity qam-4096-light {
  base coding-modulation;
  description
    "4096 QAM light coding and modulation scheme.";
}

/*
 * Protection architecture type identities
 */
identity protection-architecture-type {
  description
    "protection architecture type";
}

identity one-plus-one-type {
  base protection-architecture-type;
  description
    "One carrier termination and
    one protecting carrier termination.";
}

identity one-to-n-type {
  base protection-architecture-type;
  description
    "One carrier termination protecting
    n other carrier terminations.";
}
identity protection-states {
    description
    "Identities describing the status of the protection,
in a group of carrier terminations configured in
a radio link protection mode.";
}

identity unprotected {
    base protection-states;
    description "Not protected";
}

identity protected {
    base protection-states;
    description "Protected";
}

identity unable-to-protect {
    base protection-states;
    description "Unable to protect";
}

identity protection-external-commands{
    description
    "Protection external commands for trouble shooting
purpose.";
}

identity manual-switch{
    base protection-external-commands;
    description
    "A switch action initiated by an operator command.
It switches normal traffic signal to the protection
transport entity.";
}
/*
 * TDM-type identities
 */
identity tdm-type {
  description
    "A description of the type of TDM connection,
       also indicating the supported capacity of the
       connection."
}

identity E1 {
  base tdm-type;
  description
    "E1 connection, 2,048 Mbit/s."
}

identity STM-1 {
  base tdm-type;
  description
    "STM-1 connection, 155,52 Mbit/s."
}

/*
 * Typedefs
 */
typedef power {
  type decimal64 {
    fraction-digits 1;
  }
  description
    "Type used for power values, selected and measured.";
}

/*
 * Radio Link Terminal (RLT) - Configuration data nodes
 */
augment "/if:interfaces/if:interface" {
  when "if:type = 'mrl:radio-link-terminal'";
  description
    "Addition of data nodes for radio link terminal to
     the standard Interface data model, for interfaces of
     the type 'radio-link-terminal'.";
}
leaf id {
  type string;
  default "";
  description
    "ID of the radio link terminal. Used by far-end when checking that it’s connected to the correct RLT.";
}

leaf mode {
  type identityref {
    base rlt-mode;
  }
  mandatory true;
  description
    "A description of the mode in which the radio link terminal is configured. The format is X plus Y. X represent the number of bonded carrier terminations. Y represent the number of protecting carrier terminations.";
}

leaf-list carrier-terminations {
  type if:interface-ref;
  must "/if:interfaces/if:interface/if:name = current()" 
    + "/if:type = 'mrl:carrier-termination'" {
    description
      "The type of interface must be 'carrier-termination'.";
  }
  min-elements 1;
  description
    "A list of references to carrier terminations included in the radio link terminal.";
}

leaf-list rlp-groups {
  type leafref {
    path "/mrl:radio-link-protection-groups/
      + "mrl:radio-link-protection-group/mrl:name”;
  }
  description
    "A list of references to the carrier termination groups configured for radio link protection in this radio link terminal.";
}
leaf-list xpic-pairs {
  if-feature xpic;
  type leafref {
    path "/mrl:xpic-pairs/mrl:xpic-pair/mrl:name";
  }
  description
    "A list of references to the XPIC pairs used in this
     radio link terminal. One pair can be used by two
     terminals.";
}

leaf mimo-group {
  if-feature mimo;
  type leafref {
    path "/mrl:mimo-groups/mrl:mimo-group/mrl:name";
  }
  description
    "A reference to the MIMO group used in this
     radio link terminal. One group can be used by more
     than one terminal.";
}

list tdm-connections {
  if-feature tdm;
  key "tdm-type";
  description
    "A list stating the number of TDM connections of a
     specified tdm-type that is supported by the RLT.";
  leaf tdm-type {
    type identityref {
      base tdm-type;
    }
    description
      "The type of TDM connection, which also indicates
       the supported capacity.";
  }
  leaf tdm-connections {
    type uint16;
    mandatory true;
    description "Number of connections of the specified type.";
  }
}
augment "/if:interfaces/if:interface" {
  when "if:type = 'mrl:carrier-termination'";
  description "Addition of data nodes for carrier termination to the standard Interface data model, for interfaces of the type 'carrier-termination'.";
  leaf carrier-id {
    type string;
    default "A";
    description "ID of the carrier. (e.g. A, B, C or D) Used in XPIC & MIMO configurations to check that the carrier termination is connected to the correct far-end carrier termination. Should be the same carrier ID on both sides of the hop. Defaulted when not MIMO or XPIC.";
  }
  leaf tx-enabled {
    type boolean;
    default "false";
    description "Disables (false) or enables (true) the transmitter. Only applicable when the interface is enabled (interface:enabled = true) otherwise it’s always disabled.";
  }
  leaf tx-frequency {
    type uint32;
    units "kHz";
    mandatory true;
    description "Selected transmitter frequency.";
  }
}
leaf rx-frequency {
  type uint32;
  units "kHz";
  description
    "Selected receiver frequency.
    Mandatory and writeable when rx-frequency-config=true.
    Otherwise read-only and calculated from tx-frequency
    and duplex-distance."
}

leaf rx-frequency-config {
  type boolean;
  default "true";
  description
    "Enable (true) or disable (false) direct
    configuration of rx-frequency and instead
    using a defined duplex distance."
}

leaf duplex-distance {
  when "./.rx-frequency-config = 'false'";
  type uint32;
  units "kHz";
  mandatory true;
  description
    "Distance between Tx & Rx frequencies.
    Used to calculate rx-frequency when
    rx-frequency-config=false."
}

leaf channel-separation {
  type decimal64 {
    fraction-digits 1;
  }
  units "MHz";
  mandatory true;
  description
    "The amount of bandwidth allocated to a carrier."
}

leaf polarization {
  type enumeration {
    enum "horizontal" {
      description "Horizontal polarization."
    }
  }
}
enum "vertical" {
    description "Vertical polarization.";
}
enum "not-specified" {
    description "Polarization not specified.";
}

default "not-specified";
description
"Polarization - A textual description for info only.";

leaf power-mode {
    type enumeration {
        enum rtpc {
            description "Remote Transmit Power Control (RTPC).";
        }
        enum atpc {
            description "Automatic Transmit Power Control (ATPC).";
        }
    }
    mandatory true;
description
"A choice of Remote Transmit Power Control (RTPC)
or Automatic Transmit Power Control (ATPC).";
}

leaf selected-output-power {
    type power {
        range "-99..40";
    }
    units "dBm";
    mandatory true;
description
"Selected output power in RTPC mode and selected
maximum selected maximum output power in ATPC mode.
Minimum output power in ATPC mode is the same as the
system capability, available-min-output-power.";
}

leaf atpc-lower-threshold {
    when "/../power-mode = 'atpc'";
    type power {
        range "-99..-30";
    }
    units "dBm";
}

Ahlberg, et al.
mandatory true;
description
"The lower threshold for the input power at far-end used in the ATPC mode."
}

leaf atpc-upper-threshold {
when ".../power-mode = 'atpc'";
type power {
    range "-99..-30"
}
units "dBm";
mandatory true;
description
"The upper threshold for the input power at far-end used in the ATPC mode."
}

leaf coding-modulation-mode {
type enumeration {
    enum fixed {
        description "Fixed coding/modulation."
    }
    enum adaptive {
        description "Adaptive coding/modulation."
    }
}
mandatory true;
description
"A selection of fixed or adaptive coding/modulation mode."
}

leaf selected-cm {
when ".../coding-modulation-mode = 'fixed'"
type identityref {
    base coding-modulation;
}
mandatory true;
description
"Selected fixed coding/modulation."
}
leaf selected-min-acm {
  when "./coding-modulation-mode = 'adaptive'";
  type identityref {
    base coding-modulation;
  }
  mandatory true;
  description "Selected minimum coding/modulation. Adaptive coding/modulation shall not go below this value.";
}

leaf selected-max-acm {
  when "./coding-modulation-mode = 'adaptive'";
  type identityref {
    base coding-modulation;
  }
  mandatory true;
  description "Selected maximum coding/modulation. Adaptive coding/modulation shall not go above this value.";
}

leaf if-loop {
  type enumeration {
    enum disabled {
      description "Disables the IF Loop.";
    }
    enum client {
      description "Loops the signal back to the client side.";
    }
    enum radio {
      description "Loops the signal back to the radio side.";
    }
  }
  default "disabled";
  description "Enable (client/radio) or disable (disabled) the IF loop, which loops the signal back to the client side or the radio side.";
}
leaf rf-loop {
  type enumeration {
    enum disabled {
      description "Disables the RF Loop.";
    }
    enum client {
      description "Loops the signal back to the client side.";
    }
    enum radio {
      description "Loops the signal back to the radio side.";
    }
  }
  default "disabled";
  description "Enable (client/radio) or disable (disabled) the RF loop, which loops the signal back to the client side or the radio side."
}

container ct-performance-thresholds {
  description "Specification of thresholds for when alarms should be sent and cleared for various performance counters."

  leaf received-level-alarm-threshold {
    type power {
      range "-99..-30";
    }
    units "dBm";
    default "-99";
    description "Specification of at which received power level an alarm should be raised.";
  }

  leaf transmitted-level-alarm-threshold {
    type power {
      range "-99..40";
    }
    units "dBm";
    default "-99";
    description "An alarm is sent when the transmitted power level is below the specified threshold.";
  }
}
leaf ber-alarm-threshold {
  type enumeration {
    enum "10e-9" {
      description "Threshold at 10e-9.";
    }
    enum "10e-8" {
      description "Threshold at 10e-8.";
    }
    enum "10e-7" {
      description "Threshold at 10e-7.";
    }
    enum "10e-6" {
      description "Threshold at 10e-6.";
    }
    enum "10e-5" {
      description "Threshold at 10e-5.";
    }
    enum "10e-4" {
      description "Threshold at 10e-4.";
    }
    enum "10e-3" {
      description "Threshold at 10e-3.";
    }
    enum "10e-2" {
      description "Threshold at 10e-2.";
    }
    enum "10e-1" {
      description "Threshold at 10e-1.";
    }
  }
  default "10e-6";
  description
    "Specification of at which BER an alarm should be raised.";
}

/
  * Radio Link Terminal - Operational state data nodes
  * Currently nothing in addition to the general
  * interface-state model.
  */

/
  * Carrier Termination - Operational state data nodes
  */
augment "/if:interfaces-state/if:interface" {
  when "if:type = 'mrl:carrier-termination'";
  description "Addition of state data nodes for carrier termination to
the standard Interface state data model, for interfaces
of the type 'carrier-termination'.";

  leaf tx-oper-status {
    type enumeration {
      enum "off" {
        description "Transmitter is off.";
      }
      enum "on" {
        description "Transmitter is on.";
      }
      enum "standby" {
        description "Transmitter is in standby.";
      }
    }
    description "Shows the operative status of the transmitter.";
  }

  leaf actual-transmitted-level {
    type power {
      range "-99..40";
    }
    units "dBm";
    description "Actual transmitted power level (0.1 dBm resolution).";
  }

  leaf actual-received-level {
    type power {
      range "-99..-20";
    }
    units "dBm";
    description "Actual received power level (0.1 dBm resolution).";
  }

  leaf actual-tx-cm {
    type identityref {
      base coding-modulation;
    }
    description "Actual coding/modulation in transmitting direction.";
  }
}
leaf actual-snir {
    type decimal64 {
        fraction-digits 1;
        range "0..99";
    }
    units "dB";
    description
        "Actual signal to noise plus interference ratio. (0.1 dB resolution).";
}

leaf actual-xpi {
    if-feature xpic;
    type decimal64 {
        fraction-digits 1;
        range "0..99";
    }
    units "dB";
    description
        "The actual carrier to cross-polar interference. Only valid if XPIC is enabled. (0.1 dB resolution).";
}

container capabilities {
    description
        "Capabilities of the the installed equipment and some selected configurations.";

    leaf min-tx-frequency {
        type uint32;
        units "kHz";
        description
            "Minimum Tx frequency possible to use.";
    }

    leaf max-tx-frequency {
        type uint32;
        units "kHz";
        description
            "Maximum Tx frequency possible to use.";
    }
}
leaf min-rx-frequency {
    type uint32;
    units "kHz";
    description
        "Minimum Rx frequency possible to use.";
}

leaf max-rx-frequency {
    type uint32;
    units "kHz";
    description
        "Maximum Tx frequency possible to use.";
}

leaf available-min-output-power {
    type power;
    units "dBm";
    description
        "The minimum output power supported.";
}

leaf available-max-output-power {
    type power;
    units "dBm";
    description
        "The maximum output power supported.";
}

leaf available-min-acm {
    type identityref {
        base coding-modulation;
    }
    description
        "Minimum coding-modulation possible to use.";
}

leaf available-max-acm {
    type identityref {
        base coding-modulation;
    }
    description
        "Maximum coding-modulation possible to use.";
}
augment "/if:interfaces-state/if:interface/if:statistics" {
  when "../if:type = 'mrl:carrier-termination'";
  description
    "Addition of state data nodes in the container statistics
    for carrier terminations to the standard Interface data
    model, for interfaces of the type 'carrier-termination'.";
}

leaf bbe {
  type yang:counter32;
  units "number of block errors";
  description
    "Number of Background Block Errors (BBE) during the
    interval. A BBE is an errored block not occurring as
    part of an SES.";
}

leaf es {
  type yang:counter32;
  units "seconds";
  description
    "Number of Errored Seconds (ES) since last
    reset. An ES is a one-second period with
    one or more errored blocks or at least one
    defect.";
}

leaf ses {
  type yang:counter32;
  units "seconds";
  description
    "Number of Severely Errored Seconds (SES) during the
    interval. SES is a one-second period which contains
    equal or more than 30% errored blocks or at least
    one defect. SES is a subset of ES.";
}

leaf uas {
  type yang:counter32;
  units "seconds";
  description
    "Number of Unavailable Seconds (UAS), that is, the
    total time that the node has been unavailable during
    a fixed measurement interval.";
}
leaf min-rltm {
    type power {
        range "-99..-20";
    }
    units "dBm";
    description
        "Minimum received power level since last reset.";
}

leaf max-rltm {
    type power {
        range "-99..-20";
    }
    units "dBm";
    description
        "Maximum received power level since last reset.";
}

leaf min-tltm {
    type power {
        range "-99..40";
    }
    units "dBm";
    description
        "Minimum transmitted power level since last reset.";
}

leaf max-tltm {
    type power {
        range "-99..40";
    }
    units "dBm";
    description
        "Maximum transmitted power level since last reset.";
}

container radio-link-protection-groups {
    description
        "Configuration of radio link protected groups
        (1+1) of carrier terminations in a radio link.
        More than one protected group per radio-link-terminal
        is allowed.";
}
list radio-link-protection-group {
  key "name";
  description
  "List of protected groups of carrier terminations
  in a radio link."

  leaf name {
    type string;
    description
    "Name used for identification of the radio
    link protection group";
  }

  leaf protection-architecture-type {
    type identityref{
      base protection-architecture-type;
    }
    default "one-plus-one-type";
    description
    "The type of protection architecture
    used, e.g. one carrier termination
    protecting one carrier termination.";
  }

  leaf protection-operation-type {
    type enumeration {
      enum "non-revertive" {
        description
        "In non revertive operation, the
        traffic does not return to the
        working carrier termination if the
        switch requests are terminated. ";
      }
      enum "revertive" {
        description
        "In revertive operation, the
        traffic always returns to (or
        remains on) the working carrier
        termination if the switch requests
        are terminated. ";
      }
    }
    default "non-revertive";
    description
    "The type of protection operation, i.e.
    revertive or non-revertive operation.";
  }
}
leaf-list working-entity {
  when "../protection-operation-type = 'revertive'";
  type if:interface-ref;
  must "/if:interfaces/if:interface[if:name = current()]
    + "/if:type = 'mrl:carrier-termination'" {
    description
    "The type of a working-entity must be
     'carrier-termination'.";
  }
  min-elements 1;
  description
  "The carrier terminations over which the
  traffic normally should be transported
  over when there is no need to use the
  protecting carrier termination.";
}

leaf revertive-wait-to-restore {
  when "../protection-operation-type = 'revertive'";
  type uint16;
  units "seconds";
  default "0";
  description
  "The time to wait before switching back
  to the working carrier termination if
  protection-operation-type is revertive.";
}

leaf-list radio-link-protection-members {
  type if:interface-ref;
  must "/if:interfaces/if:interface[if:name = current()]
    + "/if:type = 'mrl:carrier-termination'" {
    description
    "The type of a protection member must
     be 'carrier-termination'.";
  }
  min-elements 2;
  description
  "Association to a group of carrier
  terminations configured for radio link
  protection and used in the radio link terminal.";
}
action protection-external-commands {
  input {
    leaf protection-external-command {
      type identityref {
        base protection-external-commands;
      }
      description "Execution of protection external commands for trouble shooting purpose.";
    }
  }
}

container xpic-pairs {
  if-feature xpic;
  description
    "Configuration of carrier termination pairs
    for operation in XPIC mode."

  list xpic-pair {
    key "name";
    description
      "List of carrier termination pairs in XPIC mode."

    leaf name {
      type string;
      description
        "Name used for identification of the XPIC pair."
    }

    leaf enabled {
      type boolean;
      default "false";
      description
        "Enable(true)/disable(false) XPIC"
    }

    leaf-list xpic-members {
      type if:interface-ref;
      must "/if:interfaces/if:interface[if:name = current()]" + "/if:type = 'mrl:carrier-termination'" {
        description
          "The type of a xpic-member must be
          'carrier-termination'."
      } min-elements 2;
      max-elements 2;
      description
        "Association to XPIC pairs used in the
        radio link terminal."
    }
  }
}
container mimo-groups {
  if-feature mimo;
  description
    "Configuration of carrier terminations
    for operation in MIMO mode.";

  list mimo-group {
    key "name";
    description
      "List of carrier terminations in MIMO mode.";

    leaf name {
      type string;
      description
        "Name used for identification of the MIMO group.";
    }

    leaf enabled {
      type boolean;
      default "false";
      description
        "Enable(true)/disable(false) MIMO";
    }

    leaf-list mimo-members {
      type if:interface-ref;
      must "/if:interfaces/if:interface[if:name = current()]" + "/if:type = 'mrl:carrier-termination'" {
        description
          "The type of a mimo-member must be
          'carrier-termination'.";
      }
      min-elements 2;
      description
        "Association to a MIMO group if used in
        the radio link terminal.";
    }
  }
}

<CODE ENDS>
5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

The security considerations of [RFC7223] also apply to this document.

6. IANA Considerations

TBD.

7. References

7.1. Normative References


7.2. Informative References


[ONF-model]


Authors’ Addresses

Jonas Ahlberg
Ericsson AB
Lindholmsgatan 11
Goeteborg 417 56
Sweden

Email: jonas.ahlberg@ericsson.com

Ye Min
Huawei Technologies Co., Ltd.
No.1899, Xiyuan Avenue
Chengdu 611731
P.R.China

Email: amy.yemin@huawei.com

Xi Li
NEC Laboratories Europe
Kurfursten-Anlage 36
69115 Heidelberg
Germany

Email: Xi.Li@neclab.eu

Koji Kawada
NEC Corporation
1753, Shimonumabe Nakahara-ku
Kawasaki, Kanagawa 211-8666
Japan

Email: k-kawada@ah.jp.nec.com

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Email: cjbc@it.uc3m.es