

CCAMP
Internet-Draft
Intended status: Standards Track
Expires: October 6, 2016

J. Ahlberg
J-O. Carlson
H-A. Lund
T. Olausson
Ericsson
April 4, 2016

Microwave Radio Link YANG Data Models
draft-ahlberg-ccamp-microwave-radio-link-00

Abstract

YANG model for managing microwave radio link functionality.

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 October 6, 2016.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	2
1.2. Tree Diagrams	2
2. Objectives	3
3. Microwave Model	3
4. Microwave Module	6
5. Acknowledgements	30
6. IANA Considerations	30
7. Security Considerations	30
8. References	30
8.1. Normative References	31
8.2. Informative References	31
Authors' Addresses	31

1. Introduction

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document is being discussed within the microwave community.

This document defines a YANG RFC 6020 [RFC6020] module for managing the radio link functionality of a microwave node. The YANG module is augmenting RFC 7223 [RFC7223] in order to model a microwave link and interface in analogy with most other data links in a router or switch.

In summary, the YANG module defined in this internet draft is: ericsson-radio-link (should be renamed to something like: if-microwave-radio-link) - Defines the model for basic configuration of a microwave radio link.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Other definitions to clarify....

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list or leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Objectives

The aim of the YANG models contained in this draft is to provide the model that is required to implement the basic radio link management of microwave devices.

This model may be augmented by vendor specific YANG modules in order to manage product specific functionality.

However, the intention is that it should not be necessary to have any vendor specific extensions to the YANG model defined in this document to implement standard microwave radio link connectivity.

3. Microwave Model

The microwave radio link model provides the parameters for managing the radio link part of a microwave node.

The "ericsson-radio-link" YANG module has the following structure:

```

module: ericsson-radio-link
  +--rw radio-link-protection-pairs!
  |   +--rw radio-link-protection-pair* [name]
  |       +--rw name string
  |       +--rw protection-switch-mode? enumeration
  |       +--rw revertive-preferred-tx? if:interface-ref
  |       +--rw revertive-wait-to-restore? uint16
  |       +--rw radio-link-protection-members* if:interface-ref
  +--ro radio-link-protection-pairs-state
  |   +--ro radio-link-protection-pair* [name]
  |       +--ro name string
  |       +--ro protection-status? enumeration
  +--rw xpic-pairs!
  |   +--rw xpic-pair* [name]

```

```

|      +--rw name                string
|      +--rw mode?               boolean
|      +--rw xplic-members*      if:interface-ref
+--rw mimo-groups!
|   +--rw mimo-group* [name]
|   |   +--rw name                string
|   |   +--rw mode?               boolean
|   |   +--rw mimo-members*      if:interface-ref
augment /if:interfaces/if:interface:
+--rw id?                        string
+--rw mode?                      enumeration
+--rw expected-far-end-id?       string
+--rw far-end-id-check?         boolean
+--rw carrier-terminations*      if:interface-ref
+--rw rlp-pairs*                 -> /radio-link-protection-pairs/radio-link-prot
ection-pair/name
+--rw xplic-pairs*               -> /xplic-pairs/xplic-pair/name
+--rw mimo-group?               -> /mimo-groups/mimo-group/name
augment /if:interfaces/if:interface:
+--rw carrier-id?                string
+--rw tx-frequency?              uint32
+--rw rx-frequency?              uint32
+--rw duplex-distance?           uint32
+--rw duplex-config?             boolean
+--rw polarization?              enumeration
+--rw power
|   +--rw (power-mode)?
|   |   +--:(RTPC)
|   |   |   +--rw selected-output-power?      int16
|   |   +--:(ATPC)
|   |   |   +--rw selected-min-output-power?   int16
|   |   |   +--rw selected-max-output-power?   int16
|   |   |   +--rw target-input-power-far-end?  int16
+--rw reference-sec?              enumeration
+--rw coding-modulation
|   +--rw (coding-modulation-mode)?
|   |   +--:(fixed)
|   |   |   +--rw selected-cm?                  coding-modulation
|   |   +--:(adaptive)
|   |   |   +--rw selected-min-acm?              coding-modulation
|   |   |   +--rw selected-max-acm?              coding-modulation
+--rw if-loop?                    boolean
+--rw rf-loop?                    boolean
+--rw ct-performance-thresholds
|   +--rw input-power-alarm-threshold?          int16
|   +--rw output-power-alarm-threshold?          int16
|   +--rw ber-alarm-threshold?                  enumeration
|   +--rw rl-1-threshold-level?                  int16
|   +--rw rlts-1-15min-set-alarm-threshold?      uint16

```

```

    +--rw rlts-1-15min-reset-alarm-threshold?  uint16
    +--rw rlts-1-24h-set-alarm-threshold?      uint32
    +--rw tl-1-threshold-level?                int16
    +--rw tlts-1-15min-set-alarm-threshold?    uint16
    +--rw tlts-1-15min-reset-alarm-threshold?  uint16
    +--rw tlts-1-24h-set-alarm-threshold?      uint32
    +--rw es-15min-set-alarm-threshold?        uint16
    +--rw es-15min-reset-alarm-threshold?      uint16
    +--rw es-24h-set-alarm-threshold?          uint32
    +--rw ses-15min-set-alarm-threshold?       uint16
    +--rw ses-15min-reset-alarm-threshold?     uint16
    +--rw ses-24h-set-alarm-threshold?         uint32
    +--rw bbe-15min-set-alarm-threshold?       uint32
    +--rw bbe-15min-reset-alarm-threshold?     uint32
    +--rw bbe-24h-set-alarm-threshold?         uint32
    +--rw acm-15min-set-alarm-threshold?       uint16
    +--rw acm-15min-reset-alarm-threshold?     uint16
    +--rw acm-24h-set-alarm-threshold?        uint32
augment /if:interfaces-state/if:interface:
    +--ro tx-oper-status?      enumeration
    +--ro actual-ouput-power?   decimal64
    +--ro actual-input-power?   decimal64
    +--ro actual-tx-cm?        coding-modulation
    +--ro actual-rx-cm?        coding-modulation
    +--ro selected-min-speed?   yang:gauge64
    +--ro selected-max-speed?   yang:gauge64
    +--ro xpic-status?         enumeration
    +--ro mimo-status?         enumeration
    +--ro actual-snr?          decimal64
    +--ro actual-xpi?          decimal64
    +--ro actual-si?           decimal64
    +--ro capabilities
        +--ro min-tx-frequency?  uint32
        +--ro max-tx-frequency?  uint32
        +--ro min-rx-frequency?  uint32
        +--ro max-rx-frequency?  uint32
        +--ro duplex-type?      enumeration
        +--ro channel-separation? decimal64
        +--ro available-min-output-power? int16
        +--ro available-max-output-power? int16
        +--ro available-min-acm?   coding-modulation
        +--ro available-max-acm?   coding-modulation
        +--ro available-min-speed? yang:gauge64
        +--ro available-max-speed? yang:gauge64
augment /if:interfaces-state/if:interface/if:statistics:
    +--ro min-rltm?  decimal64
    +--ro max-rltm?  decimal64
    +--ro min-tltm?  decimal64

```

```
+++ro max-tlrm?    decimal64
+++ro bbe?         yang:counter32
+++ro es?          yang:counter32
+++ro ses?         yang:counter32
+++ro uas?         yang:counter32
+++ro bber?        decimal64
+++ro esr?         decimal64
+++ro sesr?        decimal64
```

4. Microwave Module

This YANG module augments the interfaces defined in RFC 7223 [RFC7223].

```
<CODE BEGINS> file "ericsson-radio-link.yang"
module ericsson-radio-link {
  namespace "urn:com:ericsson:yang:common:radio-link";
  prefix rl;

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-interfaces {
    prefix if;
  }
  import iana-if-type {
    prefix ianaift;
  }

  organization
    "Ericsson AB";
  contact
    "jonas.ahlberg@ericsson.com";
  description
    "This is a module for the entities in
    a generic microwave system.";

  revision 2016-03-11 {
    description
      "Draft revision.";
    reference "";
  }

  /*
   * 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.";
}

/*
 * Typedefs
 */

typedef coding-modulation {
    type enumeration {
        enum "half-bpsk-strong";
        enum "half-bpsk";
        enum "half-bpsk-light";
        enum "bpsk-strong";
        enum "bpsk";
        enum "bpsk-light";
        enum "4-qam-strong";
        enum "4-qam";
        enum "4-qam-light";
        enum "16-qam-strong";
        enum "16-qam";
        enum "16-qam-light";
        enum "32-qam-strong";
        enum "32-qam";
        enum "32-qam-light";
        enum "64-qam-strong";
        enum "64-qam";
        enum "64-qam-light";
        enum "128-qam-strong";
        enum "128-qam";
        enum "128-qam-light";
        enum "256-qam-strong";
        enum "256-qam";
        enum "256-qam-light";
        enum "512-qam-strong";
        enum "512-qam";
        enum "512-qam-light";
        enum "1024-qam-strong";
        enum "1024-qam";
        enum "1024-qam-light";
        enum "2048-qam-strong";
```

```
    enum "2048-qam";
    enum "2048-qam-light";
    enum "4096-qam-strong";
    enum "4096-qam";
    enum "4096-qam-light";
  }
  description
    "The coding and modulation schemes supported.";
}

/*
 * Radio Link Terminal (RLT) - Configuration data nodes
 */

augment "/if:interfaces/if:interface" {
  when "if:type = 'rl: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 enumeration {
      enum "1+0";
      enum "1+1";
      enum "2+0";
      enum "2+1";
      enum "2+2";
      enum "3+0";
      enum "3+1";
      enum "3+2";
      enum "3+3";
      enum "4+0";
      enum "4+1";
      enum "4+2";
      enum "4+3";
      enum "4+4";
      enum "5+0";
      enum "5+1";
      enum "5+2";
      enum "5+3";
      enum "6+0";
    }
  }
}
```



```
    enum "6+1";
    enum "6+2";
    enum "7+0";
    enum "7+1";
    enum "8+0";
  }
  default "1+0";
  description
    "A description of the mode in which the radio link
    terminal is configured. The format is X+Y.
    X represent the number of bonded carrier terminations.
    Y represent the number of protecting carrier
    terminations.";
}
leaf expected-far-end-id {
  type string;
  default "";
  description
    "Expected ID of the radio link terminal at far-end.";
}
leaf far-end-id-check {
  type boolean;
  default "false";
  description
    "Enable(true)/disable(false) check of the ID of the
    radio link terminal at far-end. When true, the system
    verifies that far-end radio link terminal ID is equal to
    the expected. If ok then the radio link terminal status
    is UP. If not ok, then the status is stated as DOWN.";
}
leaf-list carrier-terminations {
  type if:interface-ref;
  must "/if:interfaces/if:interface[if:name = current()]"
    + "/if:type = 'carrier-termination'" {
    description
      "The type of interface must be
      'carrier-termination'.";
  }
  description
    "A list of references to carrier terminations
    included in the radio link terminal.";
}
leaf-list rlp-pairs {
  type leafref {
    path "/rl:radio-link-protection-pairs/"
      + "rl:radio-link-protection-pair/rl:name";
  }
  description
```

```
        "A list of references to the carrier termination
        pairs configured for radio link protection in this
        radio link terminal.";
    }
    leaf-list xpic-pairs {
        type leafref {
            path "/rl:xpic-pairs/rl:xpic-pair/rl: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 {
        type leafref {
            path "/rl:mimo-groups/rl:mimo-group/rl:name";
        }
        description
            "A reference to the MIMO group used in this
            radio link terminal. One group can be used by more
            than one terminal.";
    }
}

/*
 * Carrier Termination - Configuration data nodes
 */

augment "/if:interfaces/if:interface" {
    when "if:type = 'rl: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-frequency {
        type uint32;
        units "kHz";
    }
}
```

```
    default "0";
    description
        "Selected transmitter frequency.";
}
leaf rx-frequency {
    type uint32;
    units "kHz";
    default "0";
    description
        "Selected receiver frequency.
        Only writeable when duplex-config=disabled and
        duplex-type=variable.";
}
leaf duplex-distance {
    type uint32;
    units "kHz";
    default "0";
    description
        "Distance between Tx & Rx frequencies.
        Only writeable when duplex-config=true and
        duplex-type=variable.";
}
leaf duplex-config {
    type boolean;
    default "false";
    description
        "Enable (true) or disable (false) configuration of
        rxFrequency using a defined duplex distance.";
}
leaf polarization {
    type enumeration {
        enum "horizontal";
        enum "vertical";
        enum "not-specified";
    }
    default "not-specified";
    description
        "Polarization - A textual description for info only.";
}
container power {
    description
        "Container including the choice of power mode.";
    choice power-mode {
        description
            "A choice of Remote Transmit Power Control (RTPC)
            or Automatic Transmit Power Control (ATPC).";
        case RTPC {
            leaf selected-output-power {
```

```
        type int16 {
            range "-99..40";
        }
        units "dBm";
        description
            "Selected output power in RTPC mode.";
    }
}
case ATPC {
    leaf selected-min-output-power {
        type int16 {
            range "-99..40";
        }
        units "dBm";
        description
            "Selected minimum output power in
            ATPC mode.";
    }
    leaf selected-max-output-power {
        type int16 {
            range "-99..40";
        }
        units "dBm";
        description
            "Selected maximum output power in
            ATPC mode.";
    }
    leaf target-input-power-far-end {
        type int16 {
            range "-99..-30";
        }
        units "dBm";
        description
            "The wanted received input power at far-end,
            when running ATPC.";
    }
}
}
}
leaf reference-sec {
    type enumeration {
        enum "1";
        enum "2";
        enum "3";
        enum "4l";
        enum "4h";
        enum "5l";
        enum "5la";
    }
}
```

```
enum "5lb";
enum "5h";
enum "5ha";
enum "5hb";
enum "6l";
enum "6la";
enum "6lb";
enum "6h";
enum "6ha";
enum "6hb";
enum "7";
enum "7a";
enum "7b";
enum "8";
enum "8a";
enum "8b";
}
description
  "Each modulation is compliant to a Spectrum Efficiency
  Class (SEC). When running Adaptive Coding/Modulation,
  one SEC has to be selected for all the coding/
  modulations between selected-min-acm and
  selected-max-acm. This parameter is called Reference
  SEC. This setting might affect
  available-max-output-power, in order to fulfill
  spectrum requirements.";
}
container coding-modulation {
  description
    "Container including the choice of coding &
    modulation mode.";
  choice coding-modulation-mode {
    description
      "A choice of fixed or adaptive coding/modulation mode.";
    case fixed {
      leaf selected-cm {
        type coding-modulation;
        description
          "Selected fixed coding/modulation.";
      }
    }
    case adaptive {
      leaf selected-min-acm {
        type coding-modulation;
        description
          "Selected minimum coding/modulation.
          Adaptive coding/modulation shall not go
          below this value.";
      }
    }
  }
}
```

```
    }
    leaf selected-max-acm {
        type coding-modulation;
        description
            "Selected maximum coding/modulation.
            Adaptive coding/modulation shall not go
            above this value.";
    }
}
}
}
leaf if-loop {
    type boolean;
    default "false";
    description
        "Enable (true) or disable (false) the IF loop,
        which loops the signal back to the client side
        (not the radio side).";
}
leaf rf-loop {
    type boolean;
    default "false";
    description
        "Enable (true) or disable (false) the RF loop,
        which loops the signal back to the client side
        (not the radio side).";
}
container ct-performance-thresholds {
    description
        "Specification of thresholds for when alarms should
        be sent and cleared for various performance counters.";
    leaf input-power-alarm-threshold {
        type int16 {
            range "-99..-30";
        }
        units "dBm";
        default "-93";
        description
            "Specification of at which input power an alarm should
            be raised.";
    }
    leaf output-power-alarm-threshold {
        type int16 {
            range "-99..40";
        }
        units "dBm";
        description
            "An alarm is sent when the actual output power is
```

```
        below the specified threshold.";
    }
    leaf ber-alarm-threshold {
        type enumeration {
            enum "10e-9";
            enum "10e-8";
            enum "10e-7";
            enum "10e-6";
            enum "10e-5";
            enum "10e-4";
            enum "10e-3";
            enum "10e-2";
            enum "10e-1";
        }
        default "10e-6";
        description
            "Specification of at which BER an alarm should
             be raised.";
    }
    leaf rl-1-threshold-level {
        type int16 {
            range "-99..40";
        }
        units "dBm";
        description
            "Specifies the threshold level for Received Level 1.
             When the received level (input power) is below this
             level the Received Level Threshold Seconds 1 (RLTS1)
             is counted.";
    }
    leaf rlts-1-15min-set-alarm-threshold {
        type uint16 {
            range "1..900";
        }
        units "seconds";
        description
            "Specifies the threshold value in seconds for RLTS1.
             An alarm is send as soon as the received level has
             been below the Received Level 1 threshold for the
             number of seconds configured in this threshold
             during a 15 minutes interval.";
    }
    leaf rlts-1-15min-reset-alarm-threshold {
        type uint16 {
            range "1..900";
        }
        units "seconds";
        description
```

```
    "Specifies the threshold value in seconds for RLTS1.
    An alarm clearing is sent when the received level
    has been below the RLTS1 threshold less seconds than
    indicated by this threshold for a 15 minutes
    interval.";
}
leaf rlts-1-24h-set-alarm-threshold {
  type uint32 {
    range "1..86400";
  }
  units "seconds";
  description
    "Specifies the threshold value in seconds for RLTS1.
    An alarm is send as soon as the received level has
    been below the Received Level 1 threshold for the
    number of seconds configured in this threshold
    during a 24 hours interval.
    The alarm is cleared after the next 24 hours if
    the threshold is not crossed.";
}
leaf tl-1-threshold-level {
  type int16 {
    range "-100..35";
  }
  units "dBm";
  description
    "Specifies the threshold level for Transmitted
    Level 1. When the transmitted level (output power)
    is above this level the Transmitted Level Threshold
    Seconds 1 (TLTS1) is counted.";
}
leaf tlts-1-15min-set-alarm-threshold {
  type uint16 {
    range "1..900";
  }
  units "seconds";
  description
    "Specifies the threshold value in seconds for TLTS1.
    An alarm is send as soon as the transmitted level
    has been above the Transmitted Level 1 threshold
    for the number of seconds configured in this
    threshold during a 15 minutes interval.";
}
leaf tlts-1-15min-reset-alarm-threshold {
  type uint16 {
    range "1..900";
  }
  units "seconds";
}
```



```
description
  "Specifies the threshold value in seconds for TLTS1.
  An alarm clearing is sent when the transmitted level
  has been above the TLTS1 threshold less seconds than
  indicated by this threshold for a 15 minutes
  interval.";
}
leaf tlts-1-24h-set-alarm-threshold {
  type uint32 {
    range "1..86400";
  }
  units "seconds";
  description
    "Specifies the threshold value in seconds for TLTS1.
    An alarm is send as soon as the transmitted level
    has been above the Transmitted Level 1 threshold for
    the number of seconds configured in this threshold
    during a 24 hours interval.
    The alarm is cleared after the next 24 hours if
    the threshold is not crossed.";
}
leaf es-15min-set-alarm-threshold {
  type uint16 {
    range "1..900";
  }
  units "seconds";
  description
    "An alarm is sent when the number of Errored Seconds
    (es) exceeds the specified threshold in a
    15 minutes interval.";
}
leaf es-15min-reset-alarm-threshold {
  type uint16 {
    range "1..900";
  }
  units "seconds";
  description
    "An alarm clearing is sent when the the number of
    Errored Seconds (es) has been below the specified
    threshold in a 15 minutes interval.";
}
leaf es-24h-set-alarm-threshold {
  type uint32 {
    range "1..86400";
  }
  units "seconds";
  description
    "An alarm is sent when the number of Errored Seconds
```

```
        (es) exceeds the specified threshold in a
        24 hours interval.
        The alarm is cleared after the next 24 hours if
        the threshold is not crossed.";
    }
    leaf ses-15min-set-alarm-threshold {
        type uint16 {
            range "1..900";
        }
        units "seconds";
        description
            "An alarm is sent when the number of Severely
            Errored Seconds (ses) exceeds the specified
            threshold in a 15 minutes interval.";
    }
    leaf ses-15min-reset-alarm-threshold {
        type uint16 {
            range "1..900";
        }
        units "seconds";
        description
            "An alarm clearing is sent when the the number of
            Severely Errored Seconds (ses) has been below the
            specified threshold in a 15 minutes interval.";
    }
    leaf ses-24h-set-alarm-threshold {
        type uint32 {
            range "1..86400";
        }
        units "seconds";
        description
            "An alarm is sent when the number of Severely
            Errored Seconds (ses) exceeds the specified
            threshold in a 24 hours interval.
            The alarm is cleared after the next 24 hours if
            the threshold is not crossed.";
    }
    leaf bbe-15min-set-alarm-threshold {
        type uint32;
        units "number of block errors";
        description
            "An alarm is sent when the number of Background
            Block Errors (bbe) exceeds the specified threshold
            in a 15 minutes interval.";
    }
    leaf bbe-15min-reset-alarm-threshold {
        type uint32;
        units "number of block errors";
    }
```

```
description
  "An alarm clearing is sent when the the number of
    Background Block Errors (bbe) has been below the
    specified threshold in a 15 minutes interval.";
}
leaf bbe-24h-set-alarm-threshold {
  type uint32;
  units "number of block errors";
  description
    "An alarm is sent when the number of Background
      Block Errors (bbe) exceeds the specified threshold
      in a 24 hours interval.
      The alarm is cleared after the next 24 hours if
      the threshold is not crossed.";
}
leaf acm-15min-set-alarm-threshold {
  type uint16 {
    range "1..900";
  }
  units "seconds";
  description
    "An alarm is sent when time in minimum coding &
      modulation (selected-min-acm) exceeds the specified
      threshold in a 15 minutes interval.";
}
leaf acm-15min-reset-alarm-threshold {
  type uint16 {
    range "1..900";
  }
  units "seconds";
  description
    "An alarm clearing is sent when the the time in
      minimum coding & modulation (selected-min-acm)
      has been below the specified threshold in a 15 minutes
      interval.";
}
leaf acm-24h-set-alarm-threshold {
  type uint32 {
    range "1..86400";
  }
  units "seconds";
  description
    "An alarm is sent when time in minimum coding &
      modulation (selected-min-acm) exceeds the specified
      threshold in a 24 hours interval.
      The alarm is cleared after the next 24 hours if
      the threshold is not crossed.";
}
```

```
    }  
  }  
  
  /*  
  * Carrier Termination - Operational state data nodes  
  */  
  
  augment "/if:interfaces-state/if:interface" {  
    when "if:type = 'rl: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";  
        enum "on";  
        enum "standby";  
      }  
      description  
        "Shows the operative status of the transmitter.";  
    }  
    leaf actual-ouput-power {  
      type decimal64 {  
        fraction-digits 1;  
        range "-99..40";  
      }  
      units "dBm";  
      description  
        "Actual transmitted output power (0.1 dBm resolution).";  
    }  
    leaf actual-input-power {  
      type decimal64 {  
        fraction-digits 1;  
        range "-99..-20";  
      }  
      units "dBm";  
      description  
        "Actual input power (0.1 dBm resolution).";  
    }  
    leaf actual-tx-cm {  
      type coding-modulation;  
      description  
        "Actual coding/modulation in transmitting direction.";  
    }  
    leaf actual-rx-cm {  
      type coding-modulation;  
      description
```

```
        "Actual coding/modulation in receiving direction.";
    }
    leaf selected-min-speed {
        type yang:gauge64;
        units "bit/s";
        description
            "Selected minimum speed, derived from selected-min-acm
            (adaptive) or selected-cm (fixed).";
    }
    leaf selected-max-speed {
        type yang:gauge64;
        units "bit/s";
        description
            "Selected maximum speed, derived from selected-max-acm
            (adaptive) or selected-cm (fixed).";
    }
    leaf xpic-status {
        type enumeration {
            enum "locked";
            enum "unlocked";
            enum "na";
        }
        description
            "Status of the XPIC. Only valid if XPIC is enabled.";
    }
    leaf mimo-status {
        type enumeration {
            enum "locked";
            enum "unlocked";
            enum "na";
        }
        description
            "Status of the MIMO. Only valid if MIMO is enabled.";
    }
    leaf actual-snr {
        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 {
        type decimal64 {
            fraction-digits 1;
            range "0..99";
        }
    }
```

```
    }
    units "dBc";
    description
        "The actual carrier to cross-polar interference.
        Only valid if XPIC is enabled. (0.1 dBc resolution).";
}
leaf actual-si {
    type decimal64 {
        fraction-digits 1;
        range "-99..99";
    }
    units "dBc";
    description
        "Actual MIMO Spatial Interference.
        Only valid if MIMO is enabled. (0.1 dBc 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 duplex-type {
        type enumeration {
            enum "fixed";
            enum "variable";
        }
    }
}
```

```
    }
    description
        "The duplex type is given by the equipment used.
        It can be a fixed duplex distance or a variable
        distance which can be configured by selecting
        rx-frequency or duplex-distance.";
    }
    leaf channel-separation {
        type decimal64 {
            fraction-digits 1;
        }
        units "MHz";
        description
            "The amount of bandwidth allocated to a carrier.";
    }
    leaf available-min-output-power {
        type int16;
        units "dBm";
        description
            "The minimum output power supported.";
    }
    leaf available-max-output-power {
        type int16;
        units "dBm";
        description
            "The maximum output power supported.";
    }
    leaf available-min-acm {
        type coding-modulation;
        description
            "Minimum coding-modulation possible to use.";
    }
    leaf available-max-acm {
        type coding-modulation;
        description
            "Maximum coding-modulation possible to use.";
    }
    leaf available-min-speed {
        type yang:gauge64;
        units "bit/s";
        description
            "Minimum speed that can be supported given
            by the available-min-acm.";
    }
    leaf available-max-speed {
        type yang:gauge64;
        units "bit/s";
        description
```

```
        "Maximum speed that can be supported given
          by the available-max-acm.";
      }
    }
  }
  augment "/if:interfaces-state/if:interface/if:statistics" {
    when "../if:type = 'rl: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 min-rltm {
      type decimal64 {
        fraction-digits 1;
        range "-99..-20";
      }
      units "dBm";
      description
        "Minimum input power since last reset.";
    }
    leaf max-rltm {
      type decimal64 {
        fraction-digits 1;
        range "-99..-20";
      }
      units "dBm";
      description
        "Maximum input power since last reset.";
    }

    leaf min-tltm {
      type decimal64 {
        fraction-digits 1;
        range "-99..40";
      }
      units "dBm";
      description
        "Minimum output power since last reset.";
    }
    leaf max-tltm {
      type decimal64 {
        fraction-digits 1;
        range "-99..40";
      }
      units "dBm";
      description
        "Maximum output power since last reset.";
```



```
}
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 bber {
  type decimal64 {
    fraction-digits 18;
  }
  units "ratio";
  description
    "Background Block Error (BBER) ratio, that is, the
     ratio of errored blocks to total blocks during a fixed
     measurement interval. BBER does not include any blocks
     during SES and Unavailable Time.";
}
leaf esr {
  type decimal64 {
    fraction-digits 18;
```

```
    }
    units "ratio";
    description
        "Errored Seconds (ESR) ratio, that is, the ratio of
        Errored Seconds to Available Time during a fixed
        measurement interval.";
}
leaf sesr {
    type decimal64 {
        fraction-digits 18;
    }
    units "ratio";
    description
        "Severely Errored Seconds Ratio (SESR), that is, the
        ratio of SES to total seconds in Available Time during
        a fixed measurement interval.";
}
}

/*
 * Radio Link Protection Pairs - Configuration data nodes
 */

container radio-link-protection-pairs {
    presence "enables radio link protection configuration";
    description
        "Configuration of radio link protected pairs (1+1) of
        carrier terminations in a radio link. More than one
        protected pair per radio-link-terminal is allowed.";
    list radio-link-protection-pair {
        key "name";
        description
            "List of protected pairs of carrier terminations
            in a radio link.";
        leaf name {
            type string;
            description
                "Name used for identification of the radio
                link protection pair";
        }
        leaf protection-switch-mode {
            type enumeration {
                enum "manual";
                enum "auto";
                enum "auto-and-revertive";
            }
            default "auto";
            description
```

```

    "The mode in which the switching mechanism is
    configured.
        Auto - System automatically switches from the
        degraded to the other receiver(Rx) or transmitter (Tx).
        Manual - Automatic switching is disabled and
        operator can select Tx manually.
        Auto & Revertive - makes it possible to manually
        set the preferred Tx.";
}
leaf revertive-preferred-tx {
    type if:interface-ref;
    must "/if:interfaces/if:interface[if:name = current()]"
        + "/if:type = 'carrier-termination'" {
        description
            "The type of a preferred-tx must be
            'carrier-termination'.";
    }
    description
        "Association to the preferred carrier
        termination in revertive protection mode.";
}
leaf revertive-wait-to-restore {
    type uint16;
    units "seconds";
    default "0";
    description
        "The time to wait before switching back to the
        preferred Tx in Auto & Revertive mode.";
}
leaf-list radio-link-protection-members {
    type if:interface-ref;
    must "/if:interfaces/if:interface[if:name = current()]"
        + "/if:type = 'carrier-termination'" {
        description
            "The type of a protection member must be
            'carrier-termination'.";
    }
    min-elements 2;
    max-elements "2";
    description
        "Association to a pair of carrier terminations
        configured for radio link protection and used in
        the radio link terminal.";
}
}
}
/*

```

```
/*
 * Radio Link Protection Pairs - Configuration state data nodes
 */

container radio-link-protection-pairs-state {
    config false;
    description
        "State data for radio link protected pairs (1+1) of
        carrier terminations in a radio link.";
    list radio-link-protection-pair {
        key "name";
        description
            "List of protected pairs of carrier terminations
            in a radio link.";
        leaf name {
            type string;
            description
                "Name used for identification of the radio
                link protection pair.";
        }
        leaf protection-status {
            type enumeration {
                enum "unprotected";
                enum "protected";
                enum "unable-to-protect";
            }
            description
                "Status of the protection, in a pair of carrier
                terminations configured in a radio link protection
                mode.";
        }
    }
}

/*
 * XPIC & MIMO groups - Configuration data nodes
 */

container xplic-pairs {
    presence "enables xplic configuration";
    description
        "Configuration of carrier termination pairs
        for operation in XPIC mode.";
    list xplic-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 mode {
        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 = '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 {
    presence "enables mimo configuration";
    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 mode {
            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 = '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. Acknowledgements

6. IANA Considerations

This document defines a new YANG module and the authors politely request that IANA assigns unique names to the YANG module file contained within this draft, and also appropriate URIs in the "IETF XML Registry".

7. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol RFC 6241 [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory to implement secure transport is SSH RFC 6242 [RFC6242]. The NETCONF access control model RFC 6536 [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 this 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. These are the subtrees and data nodes and their sensitivity/vulnerability:

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <<http://www.rfc-editor.org/info/rfc7223>>.

8.2. Informative References

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<http://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<http://www.rfc-editor.org/info/rfc6536>>.

Authors' Addresses

Jonas Ahlberg
Ericsson
Sweden

Email: jonas.ahlberg@ericsson.com

Jan-Olof Carlson
Ericsson
Sweden

Email: jan-olof.carlson@ericsson.com

Hans-Ake Lund
Ericsson
Sweden

Email: hans-ake.lund@ericsson.com

Thomas Olausson
Ericsson
Sweden

Email: thomas.olausson@ericsson.com