

CCAMP Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

N. Sambo
P. Castoldi
A. Sgambelluri
Scuola Superiore Sant'Anna
G. Fioccola
Huawei Technologies
F. Cugini
CNIT
D. Ceccarelli
Ericsson
H. Song
T. Zhou
Huawei
October 22, 2018

Finite state machine YANG model augmentation for Transponder
Reconfiguration
draft-sambo-ccamp-yang-fsm-transponder-reconf-02

Abstract

YANG enables to compile a set of consistent vendor-neutral data models for optical networks and components based on actual operational needs emerging from heterogeneous use cases. A YANG model has been also proposed to describe finite state machine to program network elements that are modeled with YANG. This document augments the more generic YANG model for finite state machine [[I-D.sambo-netmod-yang-fsm](#)], in order to pre-instruct an optical transponder on the actions to be performed (e.g., code adaptation) in case some events, such as physical layer degradations, occur.

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 <https://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 April 25, 2019.

Internet-Draft

FSM YANG for Transponder Reconf

October 2018

Copyright Notice

Copyright (c) 2018 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 (<https://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
2.	Conventions used in this document	3
3.	Terminology	3
4.	Flexible Transponders	4
5.	Augmenting the FSM YANG model for transponder reconfiguration	7
6.	Code of the YANG model for transponder reconfiguration . . .	10
7.	Acknowledgements	16
8.	Security Considerations	16
9.	IANA Considerations	16
10.	References	16
10.1.	Normative References	16
10.2.	Informative References	16
	Authors' Addresses	17

[1.](#) Introduction

Networks are evolving toward more programmability, flexibility, and multi-vendor interoperability. Multi-vendor interoperability can be applied in the context of nodes, i.e. a node composed of components provided by different vendors (named fully disaggregated white box) is assembled under the same control system. This way, operators can optimize costs and network performance without the need of being tied to single vendor equipment. NETCONF protocol [RFC6241](#) [[RFC6241](#)] based on YANG data modeling language [RFC6020](#) [[RFC6020](#)] is emerging as a candidate Software Defined Networking (SDN) enabled protocol. First, NETCONF supports both control and management functionalities, thus

permits high programmability. Then, YANG enables data modeling in a vendor-neutral way. Some recent works have provided YANG models to describe attributes of links (e.g., identification), nodes (e.g., connectivity matrix), media channels, and transponders (e.g., supported forward error correction - FEC) of networks

([\[I-D.ietf-i2rs-yang-network-topo\]](#) [\[I-D.vergara-ccamp-flexigrid-yang\]](#) [\[I-D.zhang-ccamp-l1-topo-yang\]](#)), also including optical technologies. A YANG model [\[I-D.sambo-netmod-yang-fsm\]](#) has been also proposed to describe finite state machines (FSMs) in order to program actions based on conditions and events in YANG-described devices. Such draft mainly refers to elastic optical networks (EONs), i.e. optical networks based on flexible grid where circuits with different bandwidth requirements are switched. EONs are expected to employ flexible transponders, i.e. transponders supporting multiple bit rates, multiple modulation formats, and multiple codes. Such transponders permits the (re-) configuration of the bit rate value based on traffic requirements, as well as the configuration of the modulation format and code based on the physical characteristics of a path (e.g., quadrature phase shift keying is more robust than 16 quadrature amplitude modulation). This document augments the YANG model for FSM [\[I-D.sambo-netmod-yang-fsm\]](#) to be applied in programming reconfiguration of transponders in EONs based on physical layer conditions. In particular, the model enables a centralized remote network controller (managed by a network operator) to instruct a transponder controller about the actions to perform when certain events (e.g., failures) occur. The actions to be taken and the events can be re-programmed on the device.

[2.](#) Conventions used in this document

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](#) [\[RFC2119\]](#).

[3.](#) Terminology

ABNO: Application-Based Network Operations

BER: Bit Error Rate

EON: Elastic Optical Network

FEC: Forward Error Correction

FSM: Finite State Machine

NETCONF: Network Configuration Protocol

OAM: Operation Administration and Maintenance

SDN: Software Defined Network

YANG: Yet Another Network Generator

[4.](#) Flexible Transponders

Flexible transponders enable several parameters' configurations, through the support of multiple modulation formats, baud rate, and forward error correction (FEC) schemes. This way, transmission parameters can be (re-)configured based on the physical layer conditions. The YANG model presented in this draft enables to pre-program reconfiguration settings of data plane devices in case of changes in the physical layer conditions. In particular, soft failures can be assumed. Soft failures imply transmission performance degradation, in turns a bit error rate (BER) increase, e.g. due to the ageing of some network devices. Without loosing generality, the ABNO architecture is assumed for the control and management of EONs ([RFC7491](#) [[RFC7491](#)]). Considering the state of the art, when pre-FEC BER passes above a predefined threshold, it is expected that an alarm is sent to the OAM Handler, which communicates with the ABNO controller that may trigger an SDN controller (that could be the Provisioning Manager of ABNO [RFC7491](#) [[RFC7491](#)]) for computing new transmission parameters. The involved ABNO modules are shown in the simplified ABNO architecture of Fig. 1. Then, transponders are reconfigured. When alarms related to several connections impacted by the soft failure are generated, this procedure may be particularly time consuming. The related workflow for transponder reconfiguration is shown in Fig. 2. The proposed model enables an SDN controller to instruct the transponder about reconfiguration of new transmission parameters values if a soft failure occurs. This can be done before the failure occurs (e.g., during the connection instantiation phase or during the connection service), so that data plane devices can promptly reconfigure

themselves without querying the SDN controller to trigger an on-demand recovery. This is expected to speed up the recovery process from soft failures. The related flow chart is shown in Fig. 3.

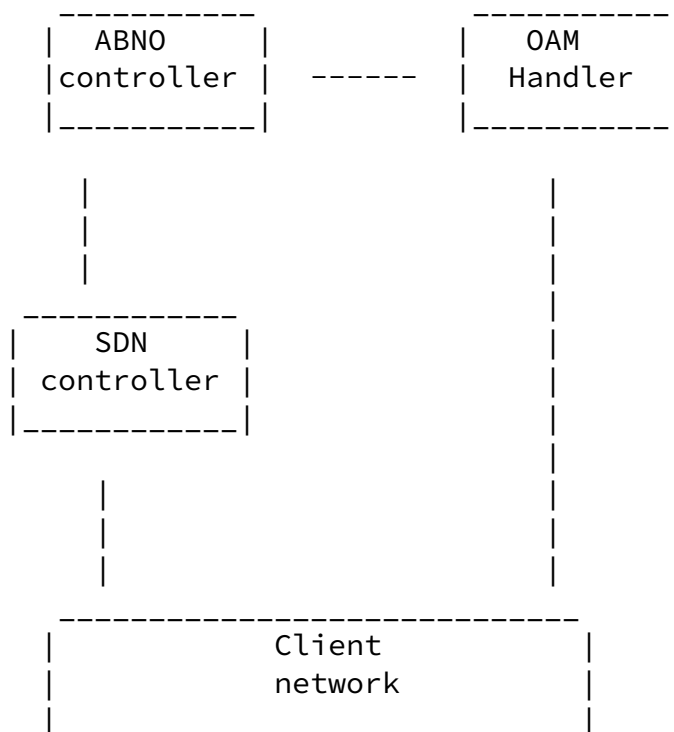
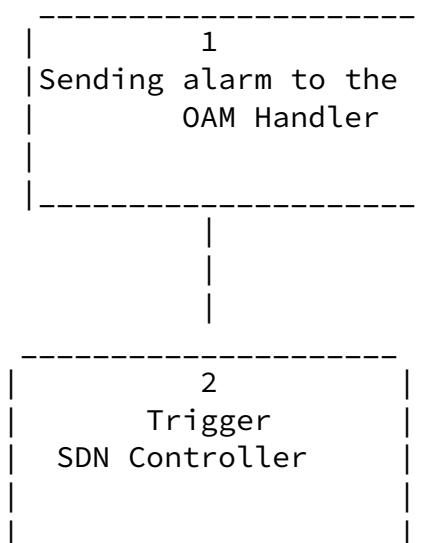


Figure 1: Assumed ABNO functional modules



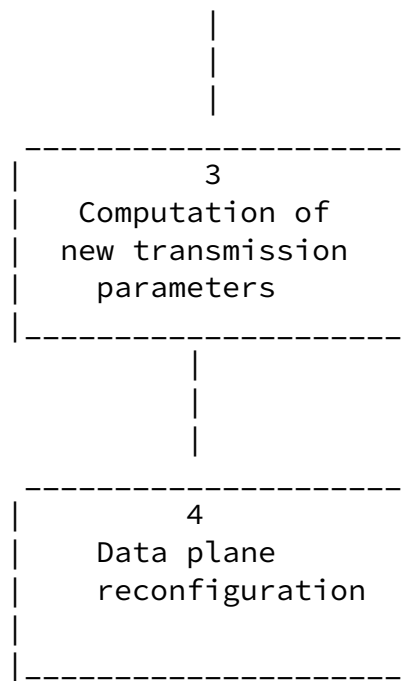
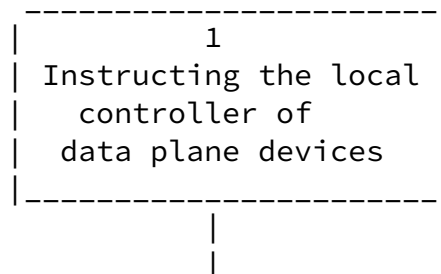


Figure 2: Flow chart of the expected state-of-the-art approach



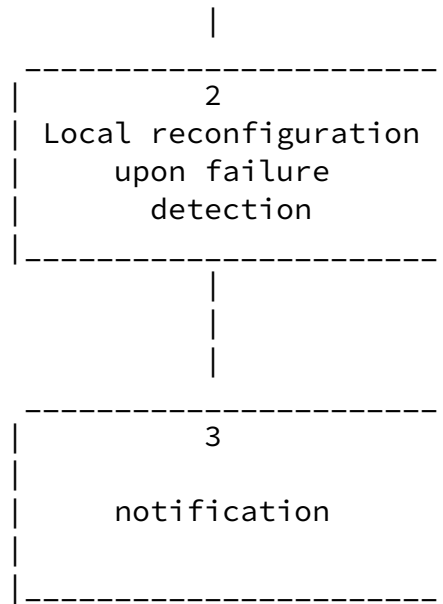


Figure 3: Flow chart of the approach exploiting YANG models in this draft

5. Augmenting the FSM YANG model for transponder reconfiguration

This section augments the FSM YANG model presented in [\[I-D.sambo-netmod-yang-fsm\]](#) to address the specific use case of transponder reconfiguration triggered by physical layer changes. The FSM is installed by the SDN controller in the local controller of the transponder and then runs there. The installation of the FSM can be enabled through a NETCONF <edit-config> message. Through FSM, the SDN controller instructs the transponder about the possible events (e.g., BER above a threshold) and reactions (e.g., change of modulation format) by setting the thresholds (e.g., BER threshold) and the reconfiguration settings. The FSM model is based on the following main attributes: states, transitions (corresponding to some specific event), and actions. In particular, more specifically with respect to [\[I-D.sambo-netmod-yang-fsm\]](#), in such a use case, a state corresponds to a specific configuration of transponder transmission parameters: e.g., given by the modulation format and the FEC. A transition is triggered when the pre-FEC BER (or another parameter such as the OSNR) is below or above a threshold. To this purpose,

with respect to [\[I-D.sambo-netmod-yang-fsm\]](#), the attribute <filter>

is expressed by the definition of thresholds and operators. The action mainly consists of the change of modulation format and/or FEC.

The Tree of the YANG model for transponder reconfiguration (augmentation of the YANG model for FSM) is reported below.

```
module: ietf-treconf
  +--rw current-state?  leafref
  +--rw states
    +--rw state [id]
      +--rw id          state-id-type
      +--rw description? string
      +--rw transitions
        +--rw transition [name]
          +--rw name          string
          +--rw description?  string
          +--rw threshold-parameter? decimal64
          +--rw threshold-operator? string
          +--rw transition-action
            +--rw action [id]
              +--rw id          transition-id-type
              +--rw type          enumeration
              +--rw simple
                +--rw execute
                +--rw next-action? transition-id-type
                +--rw next-state?  leafref
```

More specifically, the attribute <state> is a list defining all the transponder states. <transitions> is an attribute defining a list of events that may trigger the change of transponder state (e.g., BER change). <threshold-parameter> defines a threshold value, while <threshold-operator> defines the operator <,>,<=,>=. Thus, if the event BER>TH has to be modeled, the attribute <threshold-parameter> has to be set to "TH" while <threshold-operator> to ">". <actions> defines a list of actions to take during the transition (e.g. change of modulation format) <next-state> defines the next transponder state when an action is executed (e.g., new modulation format and FEC).

For more details about the other model attributes, the reader can refer to [\[I-D.sambo-netmod-yang-fsm\]](#).

In such a use case, we assume that an event (e.g., BER>TH) is revealed by the digital signal processing (DSP) of the receiver. Once the event is recognized, the modulation format and/or the FEC have to be changed, both at the receiver and the transmitter. Thus, the list of actions to be executed includes the change of

transmission parameters at the receiver side. Moreover, transmission and receiver must be synchronized about the transmission settings (modulation format and so on) for a proper transmission. Thus, when the transponder at the receiver side decides to change its state, the remote transponder at the transmitter side has to do the same state transition. To this purpose, the list of actions also includes this coordination. In particular, the transponder at the receiver side sends a message to the transmitter to synchronize about the transmission parameters to be adopted. This message can be sent over a control channel. This way both the transmitter and receiver operates with the same transmission parameters: e.g. the format, FEC, and so on.

Such transponder reconfiguration based on FSM has been successfully demonstrated by integrating control and data planes in a lab and field trials.

Finally, a last consideration concerns the impact on transmission bit rate when changing some transmission parameters. When passing from a more spectral efficient modulation format (but less robust with respect to physical impairments) to a less spectral efficient modulation format (more robust) such that could be polarization multiplexing 16 quadrature phase shift keying (PM-16QAM) and PM quadrature phase shift keying (PM-QPSK) the bit rate is reduced (halved in the case of PM-16QAM and PM-QPSK). This means that part of the traffic cannot be recovered through FSM, but needs of other restoration mechanisms (e.g., dynamic restoration). As an example, the gain of the proposed FSM mechanism promptly recovering part of the bit rate can be applied to high-priority traffic so that its recovery can be faster without involving central controller, while other classical recovery mechanisms (involving the sending of alarms, their processing, new computations and setup) can be adopted for best effort traffic (as the traffic that cannot be recovered when passing from PM-16QAM to PM-QPSK). The same happens changing the code rate: at fixed baud rate and modulation format, if the code redundancy is increased, the net bit rate is decreased. Again, part of the traffic can be promptly recovered through FSM, while the other by relying on classical recovery mechanisms. Another case of applicability is related to the "functional split" in next generation radio access networks (RANs). In this scenario, the evolved NodeB (eNB) functions are split into two new, most likely virtualized, network entities: the Central Unit (CU) deployed in centralized locations and the Distributed Unit (DU) deployed near the antenna. Several functional splits are being considered, e.g. by 3GPP in TR 38.801 and IEEE 1914 Working Group in Next Generation Fronthaul Interface (NGFI). They demand different requirements in terms of latency and capacity to the

fronthaul network connecting DU and CU. For example, in 3GPP TR 38.801, according to "Option 7c" functional split, 10.1-22.2Gb/s and

53.8-86.1Gb/s are required in the downstream and upstream links, respectively, while, according to "Option 8" functional split, 157.3Gb/s is required both in downstream and upstream links. Thus, the change of rate could reflect into a change of functional split.

6. Code of the YANG model for transponder reconfiguration

The related code is reported below.

```
<CODE BEGINS> file "ietf-treconf@2016-03-15.yang"
```

```
module ietf-treconf {
  namespace "http://sssup.it/fsm";
  prefix fsm;

  organization
    "Scuola Superiore Sant'Anna Network and Services Laboratory";

  contact
    " Editor: Matteo Dallaglio
      <mailto:m.dallaglio@sssup.it>
    ";

  description
    "This module contains a YANG definitions of a generic finite state
      machine.";

  revision 2016-03-15 {
    description "Initial Revision.";
    reference
      "RFC xxxx:";
  }

  identity TRANSITION {
    description "Base for all types of event";
  }
```

```

identity ON_CHANGE {
    base TRANSITION;
    description
        "The event when the database changes.";
}

```

```

// typedef statements

```

```

typedef transition-type {
    description "it defines the transition type";
    type identityref {
        base TRANSITION;
    }
}

typedef transition-id-type {
    description "it defines the transition id type";
    type uint32;
}

// grouping statements
grouping action-block {
    description "it defines the grouping action";
    leaf id {
        description "it defines the id of the transition";
        type transition-id-type;
    }
    leaf type {
        description "it defines if the action has to be simply executed
or if a conditional statement has to be checked before execution";

        type enumeration {
            enum "CONDITIONAL_OP"{
description "it defines the type CONDITIONAL OPERATION to check a
statement before execution. In this draft, at the moment, only SIMPLE
will be assumed";

}
            enum "SIMPLE_OP"{

```

```

description "it defines the type SIMPLE OPERATION: i.e., an operation
to be directly executed;
}
    }
    mandatory true;
}

grouping execution-top {
    description "it defines the execution attribute";
    anyxml execute {
        description "Represent the action to perform";
    }
    leaf next-action {
        type transition-id-type;
        description "the id of the next action to execute";
    }
}

```

```

}

container simple {
    when "../type = 'SIMPLE_OP'";
    description
        "Simple execution of an action without checking any condition";
    uses execution-top;
}
}

grouping action-top {
    description "it defines the grouping of action";
    list action {
        description "it defines the list of actions";
        key "id";

        ordered-by user;
        uses action-block;
    }
}

grouping on-change {
    description
        "Event occuring when a modification of one or more

```

```

        objects occurs";

    leaf threshold-parameter {
        description "it defines the threshold of an event determined by
        a threshold exceed";
        type decimal64;
    }

    leaf threshold-operator {
        description "it defines the operator to check the threshold
        exceed: <, > <=, >=";
        type string;
    }

}

grouping transition-top {
    description "it defines the grouping transition";
    leaf name {
        description "it defines the transition name";
        type string;
        mandatory true;
    }
}

```

```

    }

    leaf description {
        description "it describes the transition with a string";
        type string;
    }

    // list of all possible events
    uses on-change {
        when "type = 'ON_CHANGE'";
    }

    container transition-action {
        description "it defines the container actions to take during the
        transition";
        uses action-top;
    }
}

```

```

    }
}

grouping transitions-top {
    description "it defines the grouping transition";
    container transitions {
        description "it defines the container transitions";
        list transition {
            description "it defines the list of transitions";
            key "name";
            uses transition-top;

        }
    }
}

```

```

// data definition statements

```

```

uses transitions-top;

```

```

// extension statements

```

```

// feature statements

```

```

// augment statements

```

```

// rpc statements

```

```

// notification statements

```

```

// identity statements

```

```

// typedef statements

```

```

typedef state-id-type {
    description "it defines the id type of the states";
    type uint32;
}

```

```

// grouping statements

```



```

    type leafref {
        description "it refers to its id";
        path "../states/state/id";
    }
}

container states {
    description "it defines the container states";
    list state {
        description "it defines the list of states";
        key "id";
        uses state-top;
    }
}

// data definition statements

uses states-top;

// extension statements

// feature statements

// augment statements.

// rpc statements

// notification statements

} // module fsm

<CODE ENDS>

```

7. Acknowledgements

This work has been partially supported by the European Commission through the EU H2020 5G-TRANSFORMER Project (grant no. 761536) and the H2020 ORCHESTRA (Optical peRformanCe monitoring enabling dynamic networks using a Holistic cross-layEr, Self-configurable Truly flexible approach, grant agreement no: H2020-645360) project. The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document.

8. Security Considerations

TBD

9. IANA Considerations

TBD

10. References

10.1. 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, <<https://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, <<https://www.rfc-editor.org/info/rfc6020>>.
- [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, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", [RFC 7491](#), DOI 10.17487/RFC7491, March 2015, <<https://www.rfc-editor.org/info/rfc7491>>.

10.2. Informative References

Internet-Draft

FSM YANG for Transponder Reconf

October 2018

[I-D.ietf-i2rs-yang-network-topo]

Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A Data Model for Network Topologies", [draft-ietf-i2rs-yang-network-topo-20](#) (work in progress), December 2017.

[I-D.sambo-netmod-yang-fsm]

Sambo, N., Castoldi, P., Fioccola, G., Cugini, F., Song, H., and T. Zhou, "YANG model for finite state machine", [draft-sambo-netmod-yang-fsm-03](#) (work in progress), July 2018.

[I-D.vergara-ccamp-flexigrid-yang]

Madrid, U., Perdices, D., Lopezalvarez, V., Dios, O., King, D., Lee, Y., and G. Galimberti, "YANG data model for Flexi-Grid Optical Networks", [draft-vergara-ccamp-flexigrid-yang-06](#) (work in progress), January 2018.

[I-D.zhang-ccamp-l1-topo-yang]

zhenghaomian@huawei.com, z., Fan, Z., Sharma, A., and X. Liu, "A YANG Data Model for Optical Transport Network Topology", [draft-zhang-ccamp-l1-topo-yang-07](#) (work in progress), April 2017.

Authors' Addresses

Nicola Sambo
Scuola Superiore Sant'Anna
Via Moruzzi 1
Pisa 56124
Italy

Email: nicola.sambo@sssup.it

Piero Castoldi
Scuola Superiore Sant'Anna
Via Moruzzi 1
Pisa 56124
Italy

Email: piero.castoldi@sssup.it

Sambo, et al.

Expires April 25, 2019

[Page 17]

Internet-Draft

FSM YANG for Transponder Reconf

October 2018

Andrea Sgambelluri
Scuola Superiore Sant'Anna
Via Moruzzi 1
Pisa 56124
Italy

Email: andrea.sgambelluri@sssup.it

Giuseppe Fioccola
Huawei Technologies
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Filippo Cugini
CNIT
Via Moruzzi 1
Pisa 56124
Italy

Email: filippo.cugini@cnit.it

Daniele Ceccarelli
Ericsson
Torshamnsgatan, 48
Stockholm 164 40
Sweden

Email: daniele.ceccarelli@ericsson.com

Haoyu Song
Huawei
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: haoyu.song@huawei.com

Sambo, et al.

Expires April 25, 2019

[Page 18]

Internet-Draft

FSM YANG for Transponder Reconf

October 2018

Tianran Zhou
Huawei
156 Beiqing Road
Beijing 100095
China

Email: zhoutianran@huawei.com

