

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 17, 2019

A. Lindem
Cisco Systems
Y. Qu
Huawei
August 16, 2018

RIB YANG Data Model
draft-acee-rtgwg-yang-rib-extend-08.txt

Abstract

The Routing Information Base (RIB) is a list of routes and their corresponding administrative data and operational state.

[RFC 8349](#) defines the basic building blocks for RIB, and this model augments it to support multiple next-hops (aka, paths) for each route as well as additional attributes.

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 February 17, 2019.

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 (<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
2.	Terminology and Notation	2
2.1.	Glossary of New Terms	4
2.2.	Tree Diagrams	4
2.3.	Prefixes in Data Node Names	4
3.	Design of the Model	4
3.1.	RIB Tags and Preference	5
3.2.	Multiple next-hops	5
3.3.	Repair path	5
4.	RIB Model Tree	5
5.	RIB YANG Model	7
6.	Security Considerations	14
7.	IANA Considerations	15
8.	References	15
8.1.	Normative References	15
8.2.	Informative References	17
Appendix A.	Acknowledgments	17
	Authors' Addresses	17

[1.](#) Introduction

This document defines a YANG, [[RFC6020](#)], data model which extends the generic data model for RIB by augmenting the ietf-routing model as defined in [[RFC8349](#)].

RIB is a collection of best routes from all routing protocols. Within a protocol routes are selected based on the metrics in use by that protocol, and the protocol install its best routes to RIB. RIB selects the best route by comparing the route preference (aka, administrative distance) of the associated protocol.

The augmentations described herein extend the RIB to support multiple paths per route, route metrics, and administrative tags.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) [[RFC8342](#)].

[2.](#) Terminology and Notation

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

14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [[RFC8342](#)]:

- o client
- o server
- o configuration
- o system state
- o operational state
- o intended configuration

The following terms are defined in [[RFC7950](#)]:

- o action
- o augment
- o container
- o container with presence
- o data model
- o data node
- o feature
- o leaf
- o list
- o mandatory node
- o module
- o schema tree
- o RPC (Remote Procedure Call) operation

2.1. Glossary of New Terms

Routing Information Base (RIB): An object containing a list of routes, together with other information. See [\[RFC8349\]](#) [Section 5.2](#) for details.

2.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [\[RFC8340\]](#).

2.3. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
v4ur	ietf-ipv4-unicast-routing	[RFC8349]
v6ur	ietf-ipv6-unicast-routing	[RFC8349]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and Corresponding YANG Modules

3. Design of the Model

The YANG definitions in this document augment the ietf-routing model defined in [\[RFC8349\]](#), which provides a basis for routing system data model development. Together with modules defined in [\[RFC8349\]](#), a generic RIB Yang model is defined to implement and monitor RIB.

The models in [\[RFC8349\]](#) also define the basic configuration and operational state for both IPv4 and IPv6 static routes and this document also provides augmentations for static routes to support multiple next-hop and more next-hop attributes.

3.1. RIB Tags and Preference

Individual routes tags will be supported at both the route and next-hop level. A preference per next-hop is also supported for selection of the most preferred reachable static route.

3.2. Multiple next-hops

Both Ipv4 and IPv6 static route configuration defined in [[RFC8349](#)] have been augmented with a multi-next-hop option.

A static route/prefix can be configured to have multiple next-hops, each with their own tag and route preference.

In RIB, a route may have multiple next-hops. They can be either equal cost multiple paths (ECMP), or they may have different metrics.

3.3. Repair path

The loop-free alternate (LFA) Fast Reroute (FRR) pre-computes repair paths by routing protocols, and RIB stores the best repair path.

A repair path is augmented in RIB operation state for each path.

4. RIB Model Tree

A simplified graphical representation of the augmentations described herein is included below.

The meaning of the symbols in this diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- o Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only), "-x" RPC operations, and "-n" notifications.
- o Symbols after data node names: "?" means an optional node, "!" a container with presence, 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.


```
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rt:static-routes/v4ur:ipv4
  /v4ur:route/v4ur:next-hop/v4ur:next-hop-options
  /v4ur:simple-next-hop:
  +--rw preference?          uint32
  +--rw tag?                  uint32
  +--rw application-tag?     uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rt:static-routes/v4ur:ipv4
  /v4ur:route/v4ur:next-hop/v4ur:next-hop-options
  /v4ur:next-hop-list/v4ur:next-hop-list/v4ur:next-hop:
  +--rw preference?          uint32
  +--rw tag?                  uint32
  +--rw application-tag?     uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rt:static-routes/v6ur:ipv6
  /v6ur:route/v6ur:next-hop/v6ur:next-hop-options
  /v6ur:simple-next-hop:
  +--rw preference?          uint32
  +--rw tag?                  uint32
  +--rw application-tag?     uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rt:static-routes/v6ur:ipv6
  /v6ur:route/v6ur:next-hop/v6ur:next-hop-options
  /v6ur:next-hop-list/v6ur:next-hop-list/v6ur:next-hop:
  +--rw preference?          uint32
  +--rw tag?                  uint32
  +--rw application-tag?     uint32
augment /rt:routing/rt:ribs/rt:rib:
  +--ro rib-summary-statistics
    +--ro total-routes?      uint32
    +--ro total-active-routes? uint32
    +--ro total-route-memory? uint64
    +--ro protocol-rib-statistics* []
      +--ro rib-protocol?      identityref
      +--ro protocol-total-routes? uint32
      +--ro protocol-active-routes? uint32
      +--ro protocol-route-memory? uint64
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
  +--ro metric?              uint32
  +--ro tag?                  uint32
  +--ro application-tag?     uint32
augment /rt:routing/rt:ribs/rt:rib/rt:routes:
  +--ro repair-route* [id]
    +--ro id                  string
    +--ro next-hop
      | +--ro outgoing-interface? if:interface-state-ref
      | +--ro next-hop-address?   inet:ip-address
```



```
    +--ro metric?      uint32
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route
    /rt:next-hop/rt:next-hop-options/rt:simple-next-hop:
+--ro repair-path?
    -> /rt:routing/ribs/rib/routes/repair-route/id
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route
    /rt:next-hop/rt:next-hop-options/rt:special-next-hop:
+--ro repair-path?
    -> /rt:routing/ribs/rib/routes/repair-route/id
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route
    /rt:next-hop/rt:next-hop-options/rt:next-hop-list
    /rt:next-hop-list/rt:next-hop:
+--ro repair-path?
    -> /rt:routing/ribs/rib/routes/repair-route/id
```

5. RIB YANG Model

```
<CODE BEGINS> file "ietf-rib-extension@2018-08-16.yang"
module ietf-rib-extension {
  yang-version "1.1";
  namespace "urn:ietf:params:xml:ns:yang:ietf-rib-extension";

  prefix rib-ext;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-ipv4-unicast-routing {
    prefix "v4ur";
  }

  import ietf-ipv6-unicast-routing {
    prefix "v6ur";
  }

  organization
    "IETF RTGWG - Routing Working Group";

  contact
```


"WG Web: <<http://datatracker.ietf.org/group/rtgwg/>>
WG List: <<mailto:rtgwg@ietf.org>>

Author: Acee Lindem
<<mailto:acee@cisco.com>>
Author: Yingzhen Qu
<<mailto:yingzhen.qu@huawei.com>>;

description

"This YANG module extends the generic data model for RIB by augmenting the ietf-netmod-routing-cfg model. It is intended that the module will be extended by vendors to define vendor-specific RIB parameters.

This YANG model conforms to the Network Management Datastore Architecture (NDMA) as described in [RFC 8242](#).

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in [Section 4.c](#) of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

revision 2018-08-01 {

description

"Initial RFC Version";

reference

"RFC XXXX: A YANG Data Model for RIB Extensions.";

}

/* Groupings */

grouping rib-statistics {

description "Statistics grouping used for RIB augmentation";

container rib-summary-statistics {

config false;

description "Container for RIB statistics";

leaf total-routes {

type uint32;

description

"Total routes in the RIB from all protocols";

}


```
    leaf total-active-routes {
      type uint32;
      description
        "Total active routes in the RIB";
    }
    leaf total-route-memory {
      type uint64;
      description
        "Total memory for all routes in the RIB from all
        protocol clients";
    }
    list protocol-rib-statistics {
      description "List protocol statistics";
      leaf rib-protocol {
        type identityref {
          base rt:routing-protocol;
        }
        description "Routing protocol for statistics";
      }
      leaf protocol-total-routes {
        type uint32;
        description
          "Total number routes for protocol in the RIB";
      }
      leaf protocol-active-routes {
        type uint32;
        description
          "Number active routes for protocol in the RIB";
      }
      leaf protocol-route-memory {
        type uint64;
        description
          "Total memory for all routes in the RIB for protocol";
      }
    }
  }
}

grouping next-hop {
  description
    "Next-hop grouping";
  leaf interface {
    type if:interface-ref;
    description
      "Outgoing interface";
  }
  leaf address {
    type inet:ip-address;
```



```
        description
            "IPv4 or IPv6 Address of the next-hop";
    }
}

grouping attributes {
    description
        "Common attributes applicable to all paths";
    leaf metric {
        type uint32;
        description "Route metric";
    }
    leaf tag {
        type uint32;
        description "Route tag";
    }
    leaf application-tag {
        type uint32;
        description "Additional Application-Specific Route tag";
    }
}

grouping path-attribute {
    description
        "Path attribute grouping";
    leaf repair-path {
        type leafref {
            path "/rt:routing/rt:ribs/rt:rib/"
                + "rt:routes/repair-route/id";
        }
        description
            "IP Fast ReRoute (IPFRR) repair path, use a path
            from repair-route list";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rt:static-routes/v4ur:ipv4/"
    + "v4ur:route/v4ur:next-hop/v4ur:next-hop-options/"
    + "v4ur:simple-next-hop"
{
    description
        "Augment 'simple-next-hop' case in IPv4 unicast route.";
    leaf preference {
        type uint32;
        default "1";
        description "Route preference - Used to select among multiple
            static routes with a lower preference next-hop
            preferred and equal preference paths yielding
```



```
        Equal Cost Multi-Path (ECMP).";
    }
    leaf tag {
        type uint32;
        default "0";
        description "Route tag";
    }
    leaf application-tag {
        type uint32;
        description "Additional Application-Specific Route tag";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rt:static-routes/v4ur:ipv4/"
    + "v4ur:route/v4ur:next-hop/v4ur:next-hop-options/"
    + "v4ur:next-hop-list/v4ur:next-hop-list/v4ur:next-hop"
{
    description
        "Augment static route configuration 'next-hop-list'.";

    leaf preference {
        type uint32;
        default "1";
        description "Route preference - Used to select among multiple
            static routes with a lower preference next-hop
            preferred and equal preference paths yielding
            Equal Cost Multi-Path (ECMP).";
    }
    leaf tag {
        type uint32;
        default "0";
        description "Route tag";
    }
    leaf application-tag {
        type uint32;
        description "Additional Application-Specific Route tag";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rt:static-routes/v6ur:ipv6/"
    + "v6ur:route/v6ur:next-hop/v6ur:next-hop-options/"
    + "v6ur:simple-next-hop"
{
    description
        "Augment 'simple-next-hop' case in IPv6 unicast route.";
    leaf preference {
```



```
        type uint32;
        default "1";
        description "Route preference - Used to select among multiple
                    static routes with a lower preference next-hop
                    preferred and equal preference paths yielding
                    Equal Cost Multi-Path (ECMP).";
    }
    leaf tag {
        type uint32;
        default "0";
        description "Route tag";
    }
    leaf application-tag {
        type uint32;
        description "Additional Application-Specific Route tag";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rt:static-routes/v6ur:ipv6/"
    + "v6ur:route/v6ur:next-hop/v6ur:next-hop-options/"
    + "v6ur:next-hop-list/v6ur:next-hop-list/v6ur:next-hop"
{
    description
        "Augment static route configuration 'next-hop-list'.";

    leaf preference {
        type uint32;
        default "1";
        description "Route preference - Used to select among multiple
                    static routes with a lower preference next-hop
                    preferred and equal preference paths yielding
                    Equal Cost Multi-Path (ECMP).";
    }
    leaf tag {
        type uint32;
        default "0";
        description "Route tag";
    }
    leaf application-tag {
        type uint32;
        description "Additional Application-Specific Route tag";
    }
}

augment "/rt:routing/rt:ribs/rt:rib"
{
    description "Augment a RIB with statistics";
```



```
    uses rib-statistics;
  }

  augment "/rt:routing/rt:ribs/rt:rib/"
    + "rt:routes/rt:route"
  {
    description
      "Augment a route in RIB with tag.";
    uses attributes;
  }

  augment "/rt:routing/rt:ribs/rt:rib/"
    + "rt:routes"
  {
    description
      "Augment a route with a list of repair-paths.";
    list repair-route {
      key "id";
      description
        "A repair-path entry, which can be referenced
        by a repair-path.";
      leaf id {
        type string;
        description
          "A unique identifier.";
      }

      container next-hop {
        description
          "Route's next-hop attribute.";
        leaf outgoing-interface {
          type if:interface-state-ref;
          description
            "Name of the outgoing interface.";
        }
        leaf next-hop-address {
          type inet:ip-address;
          description
            "IP address of the next hop.";
        }
      }
    }
    leaf metric {
      type uint32;
      description "Route metric";
    }
  }
}
```



```
augment "/rt:routing/rt:ribs/rt:rib/"
  + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
  + "rt:simple-next-hop"
{
  description
    "Add more parameters to a path.";
  uses path-attribute;
}

augment "/rt:routing/rt:ribs/rt:rib/"
  + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
  + "rt:special-next-hop"
{
  description
    "Add more parameters to a path.";
  uses path-attribute;
}

augment "/rt:routing/rt:ribs/rt:rib/"
  + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
  + "rt:next-hop-list/rt:next-hop-list/rt:next-hop"
{
  description
    "This case augments the 'next-hop-options' in the routing
    model.";
  uses path-attribute;
}
}
<CODE ENDS>
```

6. Security Considerations

The YANG modules specified in this document define a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in `ietf-rib-extensions.yang` module that 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. For these augmentations to ietf-routing.yang, the ability to delete, add, and modify IPv4 and IPv6 static routes would allow traffic to be misrouted.

Some of the readable data nodes in the ietf-rib-extensions.yang module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. The exposure of the Routing Information Base (RIB) will expose the routing topology of the network. This may be undesirable since both due to the fact that exposure may facilitate other attacks. Additionally, network operators may consider their topologies to be sensitive confidential data.

All the security considerations for [RFC8349] writable and readable data nodes apply to the augmentations described herein.

7. IANA Considerations

This document registers a URI in the IETF XML registry [XML-REGISTRY]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-rib-extension

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-acl namespace: urn:ietf:params:xml:ns:yang:ietf-rib-extension prefix: ietf-rib-ext reference: RFC XXXX

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, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [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>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, [RFC 8341](#), DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 8343](#), DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", [RFC 8349](#), DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

[8.2.](#) Informative References

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", [BCP 215](#), [RFC 8340](#), DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [XML-REGISTRY]
Mealling, M., "The IETF XML Registry", [BCP 81](#), January 2004.

[Appendix A.](#) Acknowledgments

The RFC text was produced using Marshall Rose's xml2rfc tool.

The authors wish to thank Les Ginsberg, Krishna Deevi and Suyoung Yoon for their helpful comments and suggestions.

The authors wish to thank Tom Petch for review and commands.

Authors' Addresses

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
USA

EMail: acee@cisco.com

Yingzhen Qu
Huawei
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: yingzhen.qu@huawei.com

