

I2RS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 2, 2015

S. Hares  
L. Wang  
S. Zhuang  
Huawei  
July 1, 2014

An I2RS BGP Information Model  
draft-hares-i2rs-bgp-im-00.txt

## Abstract

This document introduces an information model for i2RS BGP protocol and network policy that aligns with the I2RS BGP use cases. This draft utilizes the general Policy based routing structured found in the RIB Information Model (IM) and Policy Base Policy.

## 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 January 2, 2015.

## Copyright Notice

Copyright (c) 2014 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.	Definitions and Acronyms	3
3.	BGP Data	3
3.1.	BGP Protocol	3
3.1.1.	Elements in I2RS Protocol Block	4
3.1.2.	RBNF the I2RS BGP IM Protocol Block	5
3.2.	BGP Instance List (BGP_INSTANCE_LIST)	6
3.2.1.	Elements in BGP_INSTANCE Element	6
3.2.2.	BGP_INSTANCE RBNF	9
3.3.	BGP Peer	11
3.3.1.	Reading/Writing Policy in I2RS IM model for BGP Peer	12
3.3.2.	Elements in BGP Peer	12
3.3.3.	RBNF for BGP Peer	15
3.4.	BGP Route	18
3.4.1.	BGP Local RIB	18
4.	BGP Policy	24
4.1.	BGP IM Policy RBNF	24
4.2.	RIB IM Policy RBNF	26
4.3.	BNP IM Grammar	26
5.	BGP Grammar	27
5.1.	UML grammar	28
5.2.	BGP-IM RBNF Grammar	28
6.	IANA Considerations	37
7.	Security Considerations	37
8.	Informative References	37
	Authors' Addresses	38

## 1. Introduction

The Interface to the Routing System (I2RS) provides read and write access to the information and state within the routing process within routing elements via protocol message exchange between an I2RS Client and an I2RS Agent associated with the a routing system. The [I-D.ietf-i2rs-architecture] describes the basic interactions of this exchange. One of the important functions of this messages exchange is for the I2RS client to interact with one or more I2RS agents to collect information from network routing systems.

One of the key information pieces that I2RS clients may collect from I2RS agents is the BGP information regarding BGP peers, BGP routes, and network topologies created within BGP. This document introduces a basic information model for BGP policies. This policy model can be linked with other information models such as the I2RS RIB informational model [I-D.ietf-i2rs-rib-info-model], and uses the generic policy found in the Policy Routing modules in [I-D.hares-i2rs-info-model-policy]. This basic BGP Information Model

is an extendable policy model is a product of the industry approach to I2RS.

## 2. Definitions and Acronyms

BGP: Border Gateway Protocol

Information Model: An abstract model of a conceptual domain, independent of a specific implementation or data representation

NETCONF: The Network Configuration Protocol as defined in [RFC6536]

RESTCONF: The REST-like protocol that provides a programmatic interface over HTTP for accessing data defined in YANG, using datastores defined in NETCONF Protocol [I-D.ietf-netconf-restconf] as defined in RBNF: Routing Backus-Naur Form [RFC5511].

## 3. BGP Data

BGP data includes information related to the BGP protocol instances, BGP Peer information, BGP Route information, and BGP policy. This section describes the information associated with the I2RS agent accessible via BGP.

### 3.1. BGP Protocol

BGP protocol support within a router(or virtual routing device) may have multiple instances of the BGP protocol. This section describes the information regarding the BGP protocol available to the I2RS Agent. At end of this section there is a RBNF form summarizing this topology.

#### Protocol

The BGP protocol information stores a group of information associated with a BGP protocol implementation. This group of informational includes:

- \* Router-ID (R/W) - Router-ID for BGP protocol,
- \* AS(R/W)- Autonomous System Number,
- \* protocol\_status (R) - administrative status of the protocol,
- \* shutdown\_protocol (R/W) - Shutdown the BGP protocol,
- \* ROLES (R) - a composite set of roles BGP peers play,

\* `bgp-instance-list` (R/W) - BGP protocol instances,

The section below describes these elements.

### 3.1.1. Elements in I2RS Protocol Block

This section describes each element of the I2RS Protocol Block.

#### Router-ID (R)

This I2RS BGP IM element stores the ROUTER-ID which is configured in the BGP protocol as the default router-id. Instances of a protocol may configure unique router-IDs. The router-ID is a 32 bit unsigned value imported from the running BGP process.

#### Router-ID (W)

Upon a write from the I2RS Client to this I2RS BGP IM element, this I2RS Agent will temporarily change this default in the BGP protocol. I2RS Agent-routing system software must be able to handle both the original value (from node configuration systems), and the I2RS value as valid defaults. This change in the ROUTER\_ID default is passed to the BGP instances spawned from the BGP protocol. After the I2RS Client stops the relationship with the I2RS Agent, the Router-ID value reverts to the configured value.

#### AS(R/W)

This I2RS BGP IM element stores the default AS number configured in the BGP implementation. This value is 8 byte value represented as a decimal value. This value is a read-write value. The initial value of the AS is imported by the I2RS Agent from the BGP protocol in the BGP routing system. This the normal case. If this data element is written to, I2RS stores the default AS number value in the BGP protocol as the new default AS in the BGP protocol. This does not change the default AS number in existing BGP instances with the existing peers. New BGP instances spawned after the I2RS changes the default AS value, should have the I2RS specified default AS. Upon the completion of the I2RS client-I2RS Agent association, the default AS value should be returned to the configured value.

#### `protcol_status` (R)

This I2RS BGP IM element stores an administrative status for the protocol activity of "no-shutdown" (0x00) or "shutdown" (0x01). If the routing entity the I2RS protocol is attached has BGP

"shutdown", no data will be available from any BGP instances. This is a read-only variable.

#### shutdown\_protocol(R/W)

A write of "TRUE" to the "shutdown\_protocol" status in the I2RS Agent indicates the I2RS agent should request a shutdown of the BGP protocol from the routing function. The default value is "FALSE" for this I2RS variable. The read of the "shutdown\_protocol" indicate the current status of the "shutdown\_protocol" variable in the I2RS Agent. This is different than the "protocol\_status" which indicates the status of the protocol: shutdown or not "shutdown" (running).

#### BGP\_ROLE (R)

The BGP\_ROLE provides a list of BGP Roles found within the BGP Instances associated with the BGP protocol function on the device associated with the I2RS agent. These can be ASBR (AS border router E-BGP), PE (Provider Edge E-BGP), IBGP peer, or RR (Route Reflector).

#### bgp-instance-list (R-W)

This element provides a list of BGP instances associated with the BGP routing process associated with the I2RS agent. Each BGP instance is identified by BGP\_INSTANCE\_NAME and BGP\_INSTANCE\_TYPE (AFI,SAFI tuple) Each instances contains: local-AS local-router-id, bgp\_local\_rib, bgp-peer-list described below.

### 3.1.2. RBNF the I2RS BGP IM Protocol Block

```

<bgp-protocol> ::= < ROUTER-ID >
                <AS_Number>
                <protocol-status>
                <shutdown_protocol>
                <BGP_ROLE>
                <bgp-instance-list>

<BGP_ROLE>    ::= [<ASBR >] [ <PE>]
                  [ <IBGP>] [ <RR>]

<protocol_status> ::= <SHUTDOWN> | <NO_SHUTDOWN>
<shutdown_protocol> ::= <TRUE> | <FALSE>
<bgp-instance-list> ::= (<BGP_INSTANCE> ...)

```

### 3.2. BGP Instance List (BGP\_INSTANCE\_LIST)

This section describes the list of BGP instances (BGP\_INSTANCE\_LIST). Each BGP\_INSTANCE has the following structure

- o BGP\_INSTANCE\_NAME (R/W) - Name for BGP instance
- o BGP\_INSTANCE\_CREATION\_STATUS (R/W)
- o BGP\_INSTANCE\_TYPE (R/W) - Types of BGP Instance
- o BGP\_VENDOR\_TYPE (R/W) - Vendor types
- o AFI (R/W) - address family
- o SAFI (R/W) - sub-address family
- o BGP\_MAX\_PREFIX (R/W)
- o bgp\_local\_RIB (R/W) -
- o bgp-peer-list (R/W)

#### 3.2.1. Elements in BGP\_INSTANCE Element

The definitions of the fields in BGP Instance are:

BGP\_INSTANCE\_NAME (R/W)

The BGP\_INSTANCE\_NAME provides a name to uniquely identify the BGP instance. BGP\_INSTANCE\_NAME identifies a instance with a particular AFI, SAFI, and BGP\_VENDOR\_TYPE field. A BGP instance capabilities may be summarized in the BGP\_INSTANCE\_TYPE. A write changes the BGP\_INSTANCE\_NAME, but

BGP\_INSTANCE\_CREATION(R/W)

is the flag to indicate whether the I2RS client is requesting the I2RS agent to have the BGP routing protocol create a new BGP instance for these I2RS parameters. The internal status includes:

- \* Not I2RS created (NOT-I2RS),
- \* I2RS client requests creation (I2RSCLIENT\_CREATE\_BGP\_INSTANCE),
- \* I2RS Agent rejects creation (I2RSAGENT\_REJECTS\_BGP\_INSTANCE\_CREATE),

- \* I2RS Agent attempting creation  
(I2RSAGENT\_ATTEMPTS\_BGP\_INSTANCE\_CREATE), and
- \* I2RS Agent failed creation  
(I2RSAGENT\_FAILS\_BGP\_INSTANCE\_CREATE)
- \* I2RS Agent created BGP instance  
(I2RSAGENT\_CREATED\_BGP\_INSTANCE)

#### BGP\_INSTANCE\_TYPE (R/W)

contains flag field that summarizes common AFI-SAFIs within a flag word.

- \* BGP\_IPV4\_UNI flag - is a flag for AFI of IPV4 and a SAFI of Unicast (1).
- \* BGP\_IPV4\_MULTI flag - is a flag for AFI of IPV4 and a SAFI Multicast (2).
- \* BGP\_L3VPN I is a flag for AFI of IPv4 and a SAFI of MPLS labels (4).
- \* BGP\_IPv4\_MVPN- is a flag for AFI of IPv4 and a SAFI of MCAST VPN (5).
- \* BGP\_VPLS - is a flag for an AFI of IPv4 and a SAFI VPLS (65).
- \* BGP\_IPV4\_MDT - is a flag for AFI of IPv4 and a SAFI of BGP MDT (66).
- \* BGP\_EVPN - is a flag for APFI of IPv4 and SAFI of EVPNs (70)
- \* BGP\_LS - is a flag for an AFI of IPv4 and a SAFI of BGP link state (LS) (71).
- \* BGP\_IPV4\_MPLS\_VPN - is a flag for an AFI of IPv4 and a SAFI MPLS-labeled Unicast VPNS (128).
- \* BGP\_IPv4\_MPLS\_MVPN - is a flag for an AFI of IPv4 and a SAFI MPLS-labeled Multicast VPNS (129).
- \* BGP\_IPv4\_ROUTE\_TARGET - is a flag for the AFI of IPv4 and a SAFI of Route Target Constraints (132)
- \* BGP\_IPv4\_FLOW - is a flag for an AFI of IPv4 and a SAFI of IPv4 dissemination of flow specification rules (133).

- \* BGP\_IPv4\_VPN\_FLOW - is a flag for an AFI of IPv4 and a SAFI of IPv4 VPN dissemination of flow specification rules(134).
- \* BGP\_L2AD - is a flag for an AFI of IPv4 and a SAFI L2 Auto-discovery (140)
- \* BGP\_IPv6\_UNI - is a flag for AFI of IPv6 and SAFI of Unicast (1)
- \* BGP\_IPv6\_MULTI - is a flag for API of IPv65 and a SAFI of Multicast (2)
- \* BGP\_IPV6\_MPLS\_VPN - is a flag for AFI of IPV4 and SAFI of Unicast MPLS VPNS.

#### BGP\_VENDOR\_TYPE(R/W)

contains a field to specify Vendor specific additions to identify instance.

#### bgp\_local\_RIB\_active (R/W)

contains the contains a local RIB for BGP associated with this BGP instance. Every BGP instance has its own local RIB which contains the routes received by all Peers (Adj-RIB-IN per Peer), and selected by policy as the active route by the local BGP speaker's Decision Process.

#### AFI (R/W)

The AFI value from the standard values found in IANA's Address Family Numbers.

#### SAFI (R/W)

The AFI value from the standard values found in IANA's SAFI values.

#### bgp\_local\_RIB (R/W)

contains the bgp local RIB associated with this BGP instance. Every BGP instance has its own local RIB which contains the routes received by all Peers (Adj-RIB-IN per Peer), and selected by policy as the active route by the local BGP speaker's Decision Process.

#### bgp-peer-list (R/W)



contains a list of BGP Peer information (BGP\_PEER) associated with this BGP instance. A write to this BGP Peer list adds a BGP Peer to this BGP instance.

### 3.2.2. BGP\_INSTANCE RBNF

```

<bgp-instance-list> ::= (< bgp-instance > ...)
<BGP_INSTANCE> ::= <BGP_INSTANCE_NAME>
                   <BGP_INSTANCE_CREATE>
                   <bgp-instance-type>
                   <BGP_VENDOR_TYPE>
                   <AFI>
                   <SAFI>
                   <bgp-local-rib>
                   <bgp-peer-list>

<BGP_INSTANCE_CREATE> ::= <Not-I2RS>
                          | <I2RSCLIENT_CREATE_BGP_INSTANCE>
                          | <I2RSAGENT_FAILS_BGP_INSTANCE_CREATE>
                          | <I2RSAGENT_CREATED_BGP_INSTANCE>
                          | <I2RSAGENT_BGP_INSTANCE_CREATE )>
                          | <I2RSAGENT_REJECTS_BGP_INSTANCE_CREATE>
                          | <I2RSAGENT_ATTEMPTS_BGP_INSTANCE_CREATE>
                          | <I2RSAGENT_FAILS_BGP_INSTANCE_CREATE>
                          | <I2RSAGENT_CREATED_BGP_INSTANCE>

< BGP_INSTANCE_TYPE > ::= < BGP_IPV4_UNI >
                          | < BGP_IPV4_MULTI >
                          | < BGP_IPV4_L3VPN >
                          | < BGP_IPV4_MVPN >
                          | < BGP_VPLS >
                          | < BGP_MDT >
                          | < BGP_EVPN >
                          | < BGP_LS >
                          | < BGP_IPV4_MPLS_VPN >
                          | < BGP_IPV4_MPLS_MVPN >
                          | < BGP_ROUTE_TARGET >
                          | < BGP_IPV4_FLOW >
                          | < BGP_IPV4_VPN_FLOW >
                          | < BGP_IPV4_VRF_LIST >
                          | < BGP_IPV6_VRF_LIST >
                          | < BGP_IPV6_UNI >
                          | < BGP_IPV6_VPNV6 >

<AFI> ::= <IPV4> | <IPV6>

< SAFI> ::= <UNICAST> | <MULTICAST>
           | < L3VPN > | <MPLS> | <MVPN>
           | < RT_CONSTRAIN > | < IPV4_FLOW_SPEC >
           | < VPNV4_FLOW_SPEC > | < CLNS_VPN >
           | < NG_MVPN > | < MDT_SAFI >

```

### 3.3. BGP Peer

The following are the elements in the BGP\_Peer structure

- o BGP\_PEER\_NAME (R/W)
- o Session\_address(R/W)
- o Peer\_MAX\_PREFIX (R/W)
- o BGP\_Peer\_Type
- o BGP\_Peer\_create - request to create BGP peer
- o bgp\_rib\_in (R)
- o bgp\_rib\_out (R/W)
- o Peer\_policy\_in (R/W) - Policy set name
- o Peer\_policy\_out (R/W) - Policy set name
- o Peer\_policy\_in\_pset - Policy set (R)
- o Peer\_policy\_out\_pset - Policy set (R)
- o Peer\_state\_info
  - \* Peer\_current\_state (R)
  - \* Peer\_last\_state (R)
  - \* Peer\_down\_reason (R)
  - \* Peer\_Received\_Route\_Cnt (R)
  - \* Peer\_Sent\_Route\_Cnt (R)
  - \* Peer\_Prefix\_high (R)
  - \* Peer\_prefix\_low (R)
  - \* Peer\_prefix\_ave (R)
  - \* Peer\_prefix\_time (R)
  - \* Peer\_PREFIX\_Max (R)

\* `publish_version` (R)

### 3.3.1. Reading/Writing Policy in I2RS IM model for BGP Peer

The I2RS Agent's peer inbound and outbound policies are logically considered to be stored in policy-sets as described in [I-D.hares-i2rs-info-model-policy]. Policy sets are uniquely identified by a Policy-set name/identity which the variables `Peer_policy_in` and `Peer_policy_out` refer to. The full policy set can be read from the variables `Peer_policy_in_pset` (inbound bgp peer policy), and `Peer_policy_out_pset`. A bit of explanation is in order to describe their use in the I2RS Client-I2RS Agent exchange.

Some I2RS clients will have long-term connections to I2RS Agents associated with a BGP routing function, and may choose to transfer the BGP Policy set or a set of the BGP Policy Components to the client's local memory. In this case, the I2RS Client could request the I2RS Agent to simply transfer the BGP Policy Sets name used by for the BGP Peers to minimize travel along with an indication of the last time the policy was updated.

The policy variables `Peer_policy_in` and `Peer_policy_out` allow both read and write of the Policy set IDs. The read of the `Peer_policy_in` and `Peer_policy_out` will provide simply the policy-set name. The write of the `Peer_policy_in` the I2RS agent will need to valid the the policy set name. After the policy set name is validated, the policy set link is placed in `Peer_policy_in_pset` and the BGP routing function updated with the policy set lists for the BGP Peer.

Similarly, if a write occurs to I2RS Agent's `Peer_policy_out`, the I2RS agent validates the policy-set name , replaces `Peer_policy_out_pset` with the link to the policy-set identified by the Policy-set name.

Reading either `Peer_policy_in_pset` will result in the I2RS Agent passing the policy linked to by the `Peer_policy_in` name (and linked to by `Peer_policy_in_pset`) to the I2RS Client in policy set form. Similarly, reading the `Peer_policy_out_pset` will retrieve the policy set pointed to by the `Peer_policy_out` name, and transfer the data in policy set form.

### 3.3.2. Elements in BGP Peer

The definitions of the fields in BGP peer are:

`BGP_PEER_NAME` (R/W)

This entry uniquely names a BGP Peer. If multiple TCP connections are utilized between two BGP Peers, then the BGP Peer uniquely identifies the Peering. An I2RS Write to a BGP\_PEER\_NAME changes the BGP\_PEER\_NAME For the duration of the I2RS client-agent association for the run-time BGP, but it does not change the configured BGP Peer name.

#### Peer\_Session\_Address (R/W)

provides local peer address information and remote peer address information. Please see the description below for the details of the BGP session address. A read of this information pulls this information from the BGP Peer data in the routing system. The write of this information MAY set-up an TCP session (either initial or a second in multiple TCP sessions support

#### BGP\_PEER\_TYPE (R/W)

contains the status on the type of BGP such as EBGp, IBGP, RR, RR-Client, PE, or CE. or Vendor specific types.

#### PEER\_MAX\_PREFIX(R/W)

contains the I2RS Agent's maximum prefixes allowed for this BGP Peer. A write of the PEER\_MAX\_PREFIX writes a new value into the I2RS Agent, and MAY upon configuration be read by the BGP routing function to set a new Peer\_Max\_Prefix\_run.

#### BGP\_Peer\_create(R/W)

provides the I2RS creation status for this peer as: not-created by I2RS, requested by client but not processed, or an I2RS agent's response to the request: (agent reject peer creation, agent attempting peer creation, Agent failed peer creation, and Agent created BGP Peer).

#### bgp\_rib\_in

the BGP routes received by from this BGP peer in the form of a bgp-route-list.

#### bgp\_rib\_out (R/W)

provides a a list of the routes sent to the Peer in the form of a bgp-route-list.

#### Peer\_policy\_in (R/W)

contains the policy set name for the inbound peer policy. The Information model for the policy set is defined in [I-D.hares-i2rs-info-model-policy]. The I2RS client reading a `Policy_peer_in` policy set name MAY match this against the policy-set names retrieved from the I2RS Agent. The I2RS Client writing to this `Peer_policy_in` writes a new policy name for the BGP peer policy in the I2RS agent, and signals the I2RS agent to have the bgp routing function to update the run time policy for this bgp peer for inbound traffic. This updated policy remains in the BGP Peer until either: a) I2RS Agents updates it, or b) the I2RS client-agent relationship ceases and the policy return to configured policy.

#### `Peer_policy_out` (R/W)

contains of a policy-set name for the outbound. The structure of a policy set is defined in [I-D.hares-i2rs-info-model-policy]. An I2RS client reading this policy set name should match the policy-set name to outbound policies retrieved from this I2RS Agent. A write of a new policy-set name to the I2RS agents signals the I2RS agent to validate the policy set name. After the Policy set name has been validated, the I2RS Agent SHOULD update I2rs BGP `Peer_policy_out_pset`, and signal the routing function to utilize the new policy identified by the policy-set to policy sets retrieved from the I2rs Agent. This updated policy remains in the BGP Peer until either: a) I2RS Agents updates it, or b) the I2RS client-agent relationship ceases and the BGP policy is restored to the configured policy.

#### `Peer_policy_in_pset` (R)

is the variable the I2RS client reads to receive the full policy-set used for this BGP Peers inbound policy. The I2RS Agent MUST to keep this policy variable in-sync with the I2RS Peer-Policy\_in.

#### `Peer_policy_out_pset` (R)

is the variable the I2RS client reads to receive the full policy-set used for this BGP Peers inbound policy. The I2RS Agent MUST to keep this policy variable in-sync with the I2RS Peer-Policy\_out.

#### `Peer_state_info`

provides the following readable state information copied from the BGP routing function:

- \* Peer\_current\_state (R) - the the current state of the BGP Peer in the BGP state machine.
- \* Peer\_last\_state (R) - the the last state of the BGP peer according to the BGP state machine
- \* Peer\_down\_reason (R) - the error code and sub-error code of the last peer down
- \* Peer\_received\_update\_cnt (R) - the count of the received update packets.
- \* Peer\_transmit\_update\_cnt (R) - the count of the transmit update packets.
- \* Peer\_Receive\_Route\_Cnt (R) - contains current total number of unique prefix routes received from peer
- \* Peer\_Sent\_Route\_Cnt (R) - contains current total number prefix routes sent to peer
- \* MAX\_PREFIX\_RCV\_Limit (R/W) - Maximum unique prefix routes accepted from peer
- \* MAX\_PREFIX\_XMT\_LIMIT (R/W) - Maximum unique prefix routes transmit to peer
- \* Peer\_prefix\_high (R) - contains high water mark count of total unique routes by prefix received from peer.
- \* Peer\_prefix\_low (R) - contains the low water mark count of total unique routes by prefix received from peer.
- \* Peer\_Prefix\_ave (R) - contains the average number of prefix routes from peer (running).

### 3.3.3. RBNF for BGP Peer

RBNF for the BGP Peer portion of BGP IM

```
< bgp-peer-list > ::= (< BGP_peer > ...)  
<BGP_peer> ::= <BGP_PEER_NAME>  
               <Peer_Session_Address>  
               <BGP_Peer_Type>  
               <PEER_MAX_PREFIX>  
               <BGP_Peer_create>  
               <bgp_rib_in>  
               <bgp_rib_out>
```

```

    <Peer_policy_in>
    <Peer_policy_out>
    <Peer_policy_in_pset>
    <Peer_policy_out_pset>
    [<peer_state_info>]

<BGP_PEER_NAME> ::= (alpha-numeric characters)

<Peer_Session_Address> ::= [<v4-session-address>]
    | [<v6-session-address>]
    | [<v4-port-session-address>]

< v4-session-address> ::= <LOCAL_IPV4_ADDRESS>
    <REMOTE_IPV4_ADDRESS>

<v6-session-address> ::= <LOCAL_IPV6_ADDRESS>
    <REMOTE_IPV6_ADDRESS>

<v4-port-session-addr> ::= <LOCAL_IPV4_ADDRESS>
    <LOCAL_TRANSPORT_PORT>
    <REMOTE_IPV4_ADDRESS>
    <REMOTE_TRANSPORT_PORT>

<BGP_Peer_Type> ::= [<ASBR>][<IBGP>]
    [<RR>][<RR-Client>]
    [<PE>][<CE>]
    [<BGP_VENDOR_types>]
    <BGP_VENDOR_type> ::= ...

<BGP_MAX_PREFIX> ::= INTEGER;

<BGP_Peer_create> ::= <Not_I2RS_Created_Peer> |
    <I2RSCLIENT_BGP_PEER_CREATE_REQUEST> |
    <I2RSAGENT_REJECTS_BGP_PEER_CREATE> |
    <I2RSAGENT_ATTEMPTS_BGP_INSTANCE_CREATE> |
    <I2RSAGENT_FAILS_BGP_INSTANCE_CREATE> |
    <I2RSAGENT_CREATED_BGP_PEER>

/* details on the bgp_route_list are
* contained in the bgp route section */

<bgp_rib_in > ::= <bgp-route-list> /* Adj-RIB-IN */
<bgp_rib_out > ::= <bgp-route-list> /* Adj-RIB-OUT */

/* details on the Policy_set_name and
* Policy_set are imported from
* draft-i2rs-hares-info-model-policy

```



```

*/

<bgp_policy_in> ::= <Policy_set_name>;
<bgp_policy_out> ::= <Policy_set_name>;
<bgp_policy_in_pset> ::= <Policy_set>;
<bgp_policy_out_pset> ::= <Policy_set>;

<Peer_state_info> ::=
    <Peer_current_state >
    [ <Peer_Last_state> ]
    [ <Peer_Received_Route_Cnt> ]
    [ <Peer_Send_Route_Cnt> ]
    [ <PEER_DOWN_REASON> ]
    [ <Peer_received_route_cnt> ]
    [ <Peer_sent_route_cnt> ]
    [ <Peer_prefix_high> ]
    [ <Peer_prefix_low> ]
    [ <Peer_perfix_ave> ]
    [ <Peer_prefix_time> ]
    [ <Peer_PREFIX_MAX> ]
    [ <publish_version> ]

<Peer_current_state> ::= < Established > | <IDLE>
    | < connect > | <active>
    | < open_send > | <open_confirm>

<Peer_Last_state> ::= < Established > | <IDLE>
    | < connect > | <active>
    | < open_send > | <open_confirm>

<PEER_DOWN_REASON> ::= <ERROR CODE> [ < ERROR SUB CODE> ]
<ERROR_CODE> ::= <1: Message_Header_Error>
    | <2: OPEN_Message_Error>
    | <3: UPDATE_Message_Error>
    | <4: Hold_Timer_Expired>
    | <5: Finite_State_Machine_Error >
    | <6: Cease >
    | <7: Route_Refresh_Error >

<ERROR_SUB_CODE> ::= <HEAD_ERROR_SUB_CODE>
    | <OPEN_ERROR_SUB_CODE>
    | <UPDATE_ERROR_SUB_CODE>

HEAD_ERROR_SUB_CODE> ::= <1 _ Connection_Not_Synchronized>
    | <2 _ Bad_Message_Length>
    | <3 _ Bad_Message_Type>
<OPEN_ERROR_SUB_CODE> ::= <1 _ Unsupported_Version_Number>
    | <2 _ Bad_Peer_AS>

```

```

        | <3 _ Bad_BGP_Identifier>
        | <4 _ Unsupported_Optional_Parameter>
        | <5 _ Deprecated_see_Appendix_A>
        | <6 _ Unacceptable_Hold_Time>
<UPDATE_ERROR_SUB_CODE> ::= <1_Malformed_Attribute_List>
    | <2_Unrecognized_Well_known_Attribute>
    | <3 _ Missing_Well_known_Attribute>
    | <4 _ Attribute_Flags_Error>
    | <5 _ Attribute_Length_Error>
    | <6 _ Invalid_ORIGIN_Attribute>
    | <7 _ Deprecated_see_Appendix_A>
    | <8 _ Invalid_NEXT_HOP_Attribute>
    | <9 _ Optional_Attribute_Error>
    | <10 _ Invalid_Network_Field>
    | <11 _ Malformed_AS_PATH>
<Route_Refresh_ERROR_SUB_CODE> ::= <1_Invalid_Message_Length >

<Peer_received_route_cnt> ::= COUNTER-64;
<Peer_sent_route_cnt> ::= COUNTER-64;
<Peer_prefix_high> ::= COUNTER-64;
<Peer_prefix_low> ::= COUNTER-64;
<Peer_perfix_ave> ::= COUNTER-64;
<Peer_prefix_time> ::= COUNTER-64;
<Peer_PREFIX_MAX> ::= COUNTER-64;
<publish_version> ::= COUNTER-64;

```

### 3.4. BGP Route

The routes associated with each BGP instance are the BGP local RIB, the AdjRIBIn, and the AdjRIBOut. The BGP local RIB is described below. The AdjRIBIn is the `bgp_rib_in` described in the BGP\_Peer section. The policy that sets preferences on the routes received by the peer is identified by the `Peer_policy_in` (an policy set name), and fully readable by the `Peer_policy_in_pset` (policy set pointer) entry.

The AdjRIBOut is contained in `bgp_rib_out` (a route list) described in the BGP Peer Section. The Policy associated with the AdjRIBOut is identified by the `Peer_policy_out` (a policy set name) and fully readable by the `Peer_policy_out_pset` (policy set pointer).

#### 3.4.1. BGP Local RIB

The BGP Local RIB listed below may be created and stored as a prefix tree, but described below as a `bgp_route-list` for full transmission from the I2RS Client to the I2RS Agent. The `bgp route list` has a sequence of `bgp_routes` with the following elements:

- o BGP\_ROUTE\_TYPE (R/W)
- o BGP\_ROUTE\_CREATE (R/W)
- o ROUTE\_DISTINGUISHER (R/W)
- o ip-prefix (R/W)
- o FLOW\_route (R/W)
- o EVPN\_route (R/W)
- o MVPN\_route (R/W)
- o bgp-attribute-list (R/W)
- o rt\_rib\_state\_info which includes:
  - \* rt\_rib\_current\_state
  - \* rt\_rib\_last\_state
  - \* rt\_rib\_reject\_reason
  - \* rt\_not\_preferred
- o BGP\_ROUTE\_CREATE (R/W)

#### 3.4.1.1. BGP\_LOCAL\_RIB Entries

The BGP Local RIB Entries are:

BGP\_ROUTE\_TYPE(R/W)

contains a ROUTE TYPE indicator.

bgp-route-prefix (R/W)

contains a prefix of the type specified in ROUTE\_TYPE.

bgp-attribute-list (R/W)

contains a list of bgp attributes.

bgp\_route\_creation (R/W)

provides the status of the creation of a BGP route requested by a I2RS Client to I2RS agent to be created in the BGP instance's

local route. The status includes "not-i2rs" routes for routes created by the bgp protocol function and imported into the I2RS Agent for transmission to the I2RS Client. For routes created there is a series of status values that track the I2RS Client request for BGP route creation.

bgp\_status\_info (R/W)

contains a set of BGP status that includes:

- \* rt\_rib\_current\_state
- \* rt\_rib\_last\_state
- \* rt\_rib\_reject\_reason
- \* rt\_not\_preferred

#### 3.4.1.2. BGP\_LOCAL\_RIB RBNF

The following is the RBNF for the BGP\_LOCAL\_RIB:

```

/* bgp local rib */
<bgp_local_rib> ::= <bgp-route-list>

<bgp-route-list> ::= (<bgp_route> ...)
<bgp_route> ::= <BGP_ROUTE_TYPE>
                <bgp-route-prefix>
                <bgp-attribute-list>
                <bgp_route_create>
                <bgp_rt_state_info>

<BGP_ROUTE_TYPE> ::= <BGP_ROUTE_TYPE_RD> |
                    <BGP_ROUTE_TYPE_IPV4>
                    <BGP_ROUTE_TYPE_IPV6>
                    <BGP_LABELED_IPV4>
                    <BGP_LABELED_IPV6>
                    <BGP_ROUTE_TYPE_FLOW>
                    <BGP_ROUTE_TYPE_EVPN>
                    <BGP_ROUTE_TYPE_MVPN>
                    <BGP_ROUTE_TYPE_VPLS>
                    <BGP_ROUTE_TYPE_BGP_LS>
                    <BGP_ROUTE_TYPE_VPLS>
                    <BGP_ROUTE_TYPE_L2VPN_SIGNALING_NLRI>
                    <BGP_RT_CONSTRAINT>
                    <PW_ROUTE>

<bgp-route-prefix> ::= <ipv4_route>

```

```

| <ipv6_route>
| <VPN_IPV4_route>
| <VPN_IPV6_route>
| <Labeled_IPv4>
| <Labeled_IPv6>
|   | <FLOW_route>
| <EVPN_route>           /* draft-ietf-l2vp-evpn */
| <MVPN_route>           /* RFC6514 */
| <MCAST_VPLS_route>    /* RFC7117 */
| <MDT_NLRI>             /* RFC6037 */
| <BGP_LS_route>        /* draft-ietf-idr-ls-distrib
ution */
| <VPLS_BGP_NLRI>       /* RFC4761 */
| <L2VPN_SIGNALING_NLRI> /* RFC6064 */
| <RT_CONSTRAIN>        /* RFC4684 */
| <PW_ROUTE>            /* RFC7267 */

<bgp-attribute-list> ::= [ <bgp_Origin> ]
                        [ <bgp_ASPath> ]
                        [ <bgp_Nexthop > ]
                        [ <bgp_MED> ]
                        [ <bgp_LocalPref> ]
                        [ <bgp_ATOMIC_AGGREGATE > ]
                        [ <bgp_AGGREGATOR> ]
                        [ <bgp_CommAttr > ]
                        [ <bgp_ ExtCommAttr > ]
                        [ <bgp_ ClusterList > ]
                        [ <bgp_ORIGINATOR_ID> ]

<ROUTE_DISTINGUISHER> ::= <16-bit ASN: 32-bit-Number >
                        | <32-bit IPv4_address:16-bit-Number >
                        | <32-bit ASN:16-bit Number >

<ipv4_route> ::= <IPV4_PREFIX_LENGTH> <IPV4_PREFIX>
<ipv6_route> ::= <IPV6_PREFIX_LENGTH> <IPV6_PREFIX>
<Labeled_IPv4> ::= <LENGTH> <MPLS_LABEL> <IPV4_PREFIX>
<Labeled_IPv6> ::= <LENGTH> <MPLS_LABEL> <IPV6_PREFIX>

<VPN_IPV4>   ::= <LENGTH> <MPLS_LABEL>
                  <ROUTE_DISTINGUISHER> <IPV4_PREFIX>
<VPN_IPV6>   ::= <LENGTH> <MPLS_LABEL>
                  <ROUTE_DISTINGUISHER> <IPV6_PREFIX>

<FLOW_route > ::= <flow_route_index> <acl>

<EVPN_route> ::= ( <MVPN_ROUTE_TYPE> ( <EVPN_MAC> | <EVPN_AD>
                  | <EVPN_INCLUSIVE> | <EVPN_ES> )

```

```

<EVPN_ROUTE_TYPE> ::= <Ethernet_Auto_Discovery_route>
                    | <MAC_advertisement_route>
                    | <Inclusive_Multicast_route>
                    | <Ethernet_Segment_route>

<MVPN_route> ::= <MVPN_ROUTE_TYPE> (<ipv4_prefix> | <ipv6_prefix> )
<MVPN_ROUTE_TYPE> ::= <Intra_AS_I_PMSI_AD_route>
                    | <Inter_AS_I_PMSI_AD_route>
                    | <S_PMSI_AD_route>
                    | <Leaf_AD_route>
                    | <Source_Active_AD_route>
                    | <Shared_Tree_Join_route>
                    | <Source_Tree_Join_route>
                    | <Ethernet_Auto_Discovery_route>
                    | <MAC_Advertisement_route>
                    | <Inclusive_Multicast_route>
                    | <Ethernet_Segment_route>

<MCAST_VPLS_route> ::= <MVPN_ROUTE_TYPE> <Selective_Tree_AD_route>
                    | *lt;Leaf_AD_route>

/* MDT NLRI - IPv4 address that Identifies the PE
   * that originated this route.
   * The RD identifies the VRF in that PE.
   * The group address MUST be a IPv4 multicast
   * group address and it is used to build P-tunnel.
   * All PEs attached to given MVPN must specify
   * the same group address even if the group is
   * an SSM group. MDT-SAFIs do not carry RTs, and the group
   * address is used to associated a received
   * MDT-SAFI with a VRF.
   */

<MDT_NLRI> ::= <ROUTE_DISTINGUISHER> <IPv4-address> <Group Address>

<BGPLS_ROUTE> ::= <BGPLS_ROUTE_TYPE>
                  (<NODE_NLRI> | <LINK_NLRI>
                   | <IPv4_Topology_Prefix_NLRI>
                   | <IPv6_Topology_Prefix_NLRI>)

<BGPLS_ROUTE_TYPE> ::= <NODE_NLRI_route>
                      | <LINK_NLRI_route>
                      | <IPv4_Topology_Prefix_route>
                      | <IPv6_Topology_Prefix_route>

/* VPLS_BGP_NLRI explanation
   * PE participating in the VPLS must have at least one VE ID.

```

```

* VE IDs are typically assigned by the network administrator
* with a scope local to VPLS. Therefore the VE ID should
* only belong to one PE unless the CE is multi-homed.
*
* if the PE is the VE, then it typical has one VE ID.
* if the PE is connected to several u-PEs, it has a distinct
* VE ID for each u-PE. The PE may additionally have a VE ID for
* itself, if asks acts as a VE for that VPLS.
* A label block with demultiplexer labels is
  * used to reach the VE ID.
* A VPLS BGP NLRI has a VE ID, a VE Block offset (VBO), a
* block offset size (VBS), and a Label base (LB).
* Labels for local VE are (LB+VBS-1) and for the remote
* (VBO+VBS-1).
*/

```

```

<VPLS_BGP_NLRI> ::= <ROUTE_DISTINGUISHER>
  <VE_ID> <VE_BlockOffset>
  <VE_BlockSize> <Label_Base>

```

```

<L2VPN_SIGNALING_NLRI> ::= <ROUTE_DISTINGUISHER> <PE_addr>

```

```

<RT_CONSTRAIN> ::= <origin as> <route target>

```

```

/* LENGTH of PW_route is prefix length of Route Distinguisher +
* Global_ID + Prefix + AC ID in bits
* default PW route = x length
* other: length is between 96-160 bits
*/

```

```

<PW_route> ::= <LENGTH> <ROUTE_DISTINGUISHER>
  <Global ID> <Prefix> <AC ID>

```

```

<BGP_ROUTE_CREATE> ::= <Not-I2RS_ROUTE> |
  <I2RSCLIENT_CREATE_BGP_ROUTE> |
  <I2RSAGENT_FAILS_BGP_ROUTE_CREATE> |
  <I2RSAGENT_CREATED_BGP_ROUTE> |
  <I2RSAGENT_BGP_ROUTE_CREATE)> |
  <I2RSAGENT_REJECTS_BGP_ROUTE_CREATE> |
  <I2RSAGENT_ATTEMPTS_BGP_ROUTE_CREATE> |
  <I2RSAGENT_FAILS_BGP_ROUTE_CREATE> |
  <I2RSAGENT_CREATED_BGP_ROUTE>

```

```

<bgp_state_info> ::= <rib_current_state>
  [<rib_last_state>]
  [<Rib_Rejected_REASON>]
  [<Not_Preferred>]

```

```

<rib_current_state> ::= ( < Active > | < InActive > )
                        | ( < primary > | < backup > )
                        [ < suppress > ]

<rib_last_state> ::= ( < Active > | < InActive > )
                    ( < primary > | < backup > )
                    [ < suppress > ]

<Rib_Rejected_REASON> ::= <peer_withdraw>
                          | <Policy-Results>
                          | < Nexthop_Unreachable>
                          | < Malformed_PEER_DOWN>
                          | < Malformed_DISCARD>
                          | < Malformed_IGNORE>

<NOT_PREFERRED_REASON> ::= <peer_address> | <router_id>
                          | <Cluster_list_length> | <IGP_METRIC>
                          | <peer_type> | <origin>
                          | <WEIGHT or PREFERRED_VALUE>
                          | <LOCAL_PREFERENCE>
                          | <ROUTE_TYPE>
                          | <AS_PATH_LENGTH> | <MED>

```

#### 4. BGP Policy

The BGP policy uses policy sets to define the policy for `bgp_policy_in`, `bgp_policy_out`, `bgp_policy_in_rpset`, and `bgp_policy_out_rpset`. Policy sets are defined in `[I-D.hares-i2rs-info-model-policy]`.

Policy sets contain informational model definitions for Policy groups, Policy Rules, PBR IM, local-config IM. The first section below contains the RBNF for the Policy sets, Policy Groups, and Policy Rules. The second section below contains the RBNF for the BGP Policy. The first section below contains the RBNF for these models that is utilized by the BGP Info-Model (IM).

##### 4.1. BGP IM Policy RBNF

The BGP IM policy has policy for:

- o BGP IM Tree Matches for Roles
- o Policy-Rule Extensions for Matching

```
/* Role Tree Matches */
```



```

<BGP-IM_READ_list> ::= [<BGP-IM-Tree-Match ...>]
<BGP-IM_WRITE_list> ::= [<BGP-IM-Tree-Match ...>]
<BGP-IM-Tree-Match> ::= <BGP-IM-Tree-Match-protocol-instance>
<BGP-IM-Match-Protocol-instance> ::= (<BGP_protocol> ...)

/* Policy Rule Extensions for BGP */
<Policy-Rule_Match_Nodes_BGP-IM> ::=
    (<Policy-Rule_Match_node_BGP-IM> ...)
<Policy-Rule_Match_Values_BGP-IM> ::=
    (<Policy-Rule_Match_value_BGP-IM> ...)
<Policy-Rule_Match_operators_BGP-IM> ::=
    (<Policy-Rule_operators_BGP-IM> ...)

<Policy-Rule_match_node_BGP-IM> ::= [<Policy-Rule_Match_acl> ]
    [<as_path_filter>]
    [<community_filter>]
    [<MED>]
    [<extcommunity_filter >]
    [<interface >]
    [<ipv4-route-list>]
    [<ipv6_rotue_list>]
    [<peer-address-list>]
    [<mpls_label >]
    [<rd_filter >]
    [<tag >]

<Policy-Rule_Action_Vars_BGP-IM> ::= [<adc_path_list>]
    [<adc_comm_list>]
    [comm_mod_args]
<acl> ::= <Policy-Rule_match

<peer-address-list> ::= (Peer_Session_Address ...);
<ipv4-route-list> ::= (<ipv4_route> ..)
<ipv6-route-list> ::= (<ipv6_route> ..)

<BGP_adc_path_list>:: =    [<add_as_path_list> |
    <delete_as_path_list>
    <modify_as_path_list>]

<adc_comm_list>::=    [<add_community_list > |          /* community lis
t/
    [ <delete_community_list >|
    <modify_community_list >
    <modify_MED >]

<comm_mod_args>::=    [<modify_next_hop >]
    [<modify_local_preference >]
    [<modify_mpls_label >]

```

```
[<modify_origin >]
[<modify_ip_dscp >]
[<modify_preference >]
[<modify_bgp_preference_value>]
[<modify_bgp_preference_weight>]
```

#### 4.2. RIB IM Policy RBNF

```
<RIB-IM_READ_list> ::= [<RIB-IM-Tree-Match ...>]
<RIB-IM_WRITE_list> ::= [<RIB-IM-Tree-Match ...>]
<RIB-IM-Tree-Match> ::= <RIB-IM-Match-routing-instance>
    <RIB-IM-Match-interface-list>
    <RIB-IM-Match-rib_list>
    <RIB-IM-match-route-list>;
```

#### 4.3. BNP IM Grammar

```

/* A few definitions imported from the policy set */
<policy_set> ::= <Policy_Set_Name> (<Policy_Group> ...)
<Policy_Group> ::= <Policy_Group_Identity>
                  <Policy_Group_Role>
                  <Policy_Group_priority>
                  <Policy_Group_precedence>
                  (<Policy-Rule>....);
                  [<Supporting-Policy_Group>]
                  [<Policy-Group-Extensions>]

<Policy_Rule> ::= <Policy-Rule_identity>
                  <Policy-Rule_priority>
                  <Policy-Rule_precedence>
                  <Policy-Rule_Roles>
                  (<Policy-Rule_Condition>
                  <Policy-Rule_Action> ...)
                  <Policy-Rule_Security_model>
                  <Policy-Rule_Result>
                  [<Policy_Rule_rule_extensions>]

<Policy_Result> ::= <Policy_set_name> <Policy_Group_Identifier>
                  <Policy_Rule_ID> <Policy_Action_Identifier>
                  <Policy-Rule_Result> <Policy-Rule_Action_res

ult>

<Policy-Rule_identity> ::= <Rule-Name> [<Rule-Secure-Identity>]
<Policy-Rule_priority> ::= INTEGER (0..500);
<Policy-Rule_precedence> ::= INTEGER (0..500);
<Policy-Rule_Roles> ::= (<Policy_Rule_Role> ...);
<Policy-Rule_Condition> ::= <Policy-Rule_node>
                          (<Policy-Rule_value> ...)
                          <Policy-Rule_mode>
                          [<Policy_Rule_Match_Operator>]
                          [<Policy_Rule_Condition_extension>]

<Policy-Rule_Role> ::= <RESOURCES> | <Policy_Rule_Scope>
<RESOURCES> ::= [<I2RS_AGENT_RESOURCE>]

```

## 5. BGP Grammar

This section examines the grammar.

## 5.1. UML grammar

This will include a UML version of the grammar.

## 5.2. BGP-IM RNBF Grammar

```

<BGP_protocol> ::= < Router ID >
                  <AS Number>
                  <protocol_status>
                  <shutdown_protocol>
                  <BGP_ROLE>
                  <bgp-instance-list>

<BGP_ROLE>      ::= [<ASBR >] [ <PE>]
                  [ < IBGP >] [ < RR >]

<protocol_status> ::= <SHUTDOWN> | <NO_SHUTDOWN>
<shutdown_protocol> ::= <TRUE> | <FALSE>
<bgp-instance-list> ::= (< BGP_INSTANCE > ...)

/* BGP Instance */
<bgp-instance-list> ::= (< BGP_INSTANCE > ...)
<BGP_INSTANCE> ::= <BGP_INSTANCE_NAME>
                  <BGP_INSTANCE_CREATE>
                  <BGP_INSTANCE_TYPE>
                  <BGP_VENDOR_TYPE>
                  <AFI>
                  <SAFI>
                  <bgp_local_rib>
                  <bgp-peer-list>

<BGP_INSTANCE_CREATE> ::= <Not-I2RS>
                        | <I2RSCLIENT_CREATE_BGP_INSTANCE>
                        | <I2RSAGENT_FAILS_BGP_INSTANCE_CREATE>
                        | <I2RSAGENT_CREATED_BGP_INSTANCE>
                        | <I2RSAGENT_BGP_INSTANCE_CREATE>
                        | <I2RSAGENT_REJECTS_BGP_INSTANCE_CREATE>
                        | <I2RSAGENT_ATTEMPTS_BGP_INSTANCE_CREATE>
                        | <I2RSAGENT_FAILS_BGP_INSTANCE_CREATE>
                        | <I2RSAGENT_CREATED_BGP_INSTANCE>

< BGP_INSTANCE_TYPE > ::= < BGP_IPV4_UNI >
                        | < BGP_IPV4_MULTI >
                        | < BGP_IPV4_L3VPN >
                        | < BGP_IPV4_MVPN >
                        | < BGP_VPLS >
                        | < BGP_MDT >

```

```

| < BGP_EVPN >
| < BGP_LS >
| < BGP_IPV4_MPLS_VPN >
| < BGP_IPV4_MPLS_MVPN >
| < BGP_Route_Target >
| < BGP_IPV4_FLOW >
| < BGP_IPV4_VPN_FLOW >
| < BGP_IPV4_VRF_LIST >
| < BGP_IPV6_VRF_LIST >
| < BGP_IPV6_UNI >
| < BGP_IPV6_VPNV6 >

<AFI> ::= <IPV4> | <IPV6>

< SAFI> ::= <UNICAST> | <MULTICAST>
| < L3VPN > | <MPLS> | <MVPN>
| < RT_CONSTRAIN > | <IPV4_FLOW_SPEC >
| < VPNV4_FLOW_SPEC > | <CLNS_VPN >
| < NG_MVPN > | < MDT_SAFI >

/* BGP Peer information */

<bgp-peer-list> ::= (< BGP_peer > ...)

<BGP_peer> ::= <BGP_PEER_NAME>
<Peer_Session_Address>
<BGP_PEER_TYPE>
<PEER_MAX_PREFIX>
<BGP_Peer_create>
<bgp_rib_in>
<bgp_rib_out>
<Peer_policy_in>
<Peer_policy_out>
<Peer_policy_in_pset>
<Peer_policy_out_pset>
[<peer_state_info>]

<BGP_PEER_NAME> ::= (alpha-numeric characters)

<Peer_Session_Address> ::= [<v4-session-address>]
| [<v6-session-address>]
| [<v4-port-session-address>]

<v4-session-address> ::= <LOCAL_IPV4_ADDRESS>
<REMOTE_IPV4_ADDRESS>

<v6-session-address> ::= <LOCAL_IPV6_ADDRESS>

```

```

    <REMOTE_IPV6_ADDRESS>

<v4-port-session-addr> ::= <LOCAL_IPV4_ADDRESS>
    <LOCAL_TRANSPORT_PORT>
    <REMOTE_IPV4_ADDRESS>
    <REMOTE_TRANSPORT_PORT>

<BGP_Peer_Type> ::= [ <ASBR> ] [ <IBGP> ]
    [ <RR> ] [ <RR-Client> ]
    [ <PE> ] [ <CE> ]
    [ <BGP_VENDOR_TYPES> ]
    <BGP_VENDOR_TYPE> ::= ...

<BGP_MAX_PREFIX> ::= INTEGER;

<BGP_Peer_create> ::= <Not_I2RS_Created_Peer> |
    <I2RSCLIENT_BGP_PEER_CREATE_REQUEST> |
    <I2RSAGENT_REJECTS_BGP_PEER_CREATE> |
    <I2RSAGENT_ATTEMPTS_BGP_INSTANCE_CREATE> |
    <I2RSAGENT_FAILS_BGP_INSTANCE_CREATE> |
    <I2RSAGENT_CREATED_BGP_PEER>

/* details on the bgp_route_list are
* contained in the bgp route section */

<bgp_rib_in > ::= <bgp-route-list> /* Adj-RIB-IN */
<bgp_rib_out > ::= <bgp-route-list> /* Adj-RIB-OUT */

/* details on the Policy_set_name and
* Policy_set are imported from
* draft-i2rs-hares-info-model-policy
*/

<bgp_policy_in> ::= <Policy_set_name>;
<bgp_policy_out> ::= <Policy_set_name>;
<bgp_policy_in_pset> ::= <Policy_set>;
<bgp_policy_out_pset> ::= <Policy_set>;

    <Peer_state_info> ::=    <Peer_current_state >
                            [ <Peer_Last_state> ]
                            [ <Peer_Received_Route_Cnt> ]
                            [ <Peer_Send_Route_Cnt> ]
                            [ <PEER_DOWN_REASON> ]
                            [ <Peer_received_route_cnt> ]
                            [ <Peer_sent_route_cnt> ]
                            [ <Peer_prefix_high> ]
                            [ <Peer_prefix_low> ]

```

```

        [<Peer_prefix_ave>]
        [<Peer_prefix_time>]
        [<Peer_PREFIX_MAX>]
        [<publish_version>]

<Peer_current_state> ::= < Established > | <IDLE>
                        | < connect > | <active>
                        | < open_send > | <open_confirm>

<Peer_Last_state> ::= < Established > | <IDLE>
                      | < connect > | <active>
                      | < open_send > | <open_confirm>

<PEER_DOWN_REASON> ::= <ERROR_CODE> [< ERROR_SUB_CODE>]
<ERROR_CODE> ::= <1: Message Header Error>
                | <2: OPEN Message Error>
                | <3: UPDATE Message Error>
                | <4: Hold Timer Expired>
                | <5: Finite State Machine Error >
                | <6: Cease >
                | <7: Route Refresh>

<ERROR_SUB_CODE> ::= <HEAD_ERROR_SUB_CODE>
                   | <OPEN_ERROR_SUB_CODE>
                   | <UPDATE_ERROR_SUB_CODE>

<HEAD_ERROR SUB CODE> ::= < 1 _ Connection Not Synchronized>
                        | <2 _ Bad Message Length>
                        | <3 _ Bad Message Type>

<OPEN_ERROR_SUB_CODE> ::= <1 _ Unsupported Version Number>
                        | <2 _ Bad Peer AS>
                        | <3 _ Bad BGP Identifier>
                        | <4 _ Unsupported Optional Parameter>
                        | <5 _ [Deprecated _ see Appendix A] >
                        | <6 _ Unacceptable Hold Time>

<UPDATE_ERROR_SUB_CODE> ::= <1 _ Malformed Attribute List>
                        | <2 _ Unrecognized Well_known Attribute>
                        | <3 _ Missing Well_known Attribute>
                        | <4 _ Attribute Flags Error>
                        | <5 _ Attribute Length Error>
                        | <6 _ Invalid ORIGIN Attribute>
                        | <7 _ [Deprecated _ see Appendix A] >
                        | <8 _ Invalid NEXT_HOP Attribute>
                        | <9 _ Optional Attribute Error>
                        | <10 _ Invalid Network Field>

```

```

    | <11 _ Malformed AS_PATH>

<Route_Refresh_ERROR_SUB_CODE> ::= <1_Invalid_Message_Length >

<Peer_received_route_cnt> ::= COUNTER-64;
<Peer_sent_route_cnt> ::= COUNTER-64;
<Peer_prefix_high> ::= COUNTER-64;
<Peer_prefix_low> ::= COUNTER-64;
<Peer_perfix_ave> ::= COUNTER-64;
<Peer_prefix_time> ::= COUNTER-64;
<Peer_PREFIX_MAX> ::= COUNTER-64;
<publish_version> ::= COUNTER-64;

/* bgp local rib */
<bgp_local_rib> ::= <bgp-route-list>

<bgp-route-list> ::= (<bgp_route> ...)
<bgp_route> ::= <BGP_ROUTE_TYPE>
                <bgp-route-prefix>
                <bgp-attribute-list>
                <bgp_route_create>
                <bgp_rt_state_info>

<BGP_ROUTE_TYPE> ::= <BGP_ROUTE_TYPE_RD> |
                    | <BGP_ROUTE_TYPE_IPV4>
                    | <BGP_ROUTE_TYPE_IPV6>
                    | <BGP_LABELED_IPV4>
                    | <BGP_LABELED_IPV6>
                    | <BGP_ROUTE_TYPE_FLOW>
                    | <BGP_ROUTE_TYPE_EVPN>
                    | <BGP_ROUTE_TYPE_MVPN>
                    | <BGP_ROUTE_TYPE_VPLS>
                    | <BGP_ROUTE_TYPE_BGP_LS>
                    | <BGP_ROUTE_TYPE_VPLS>
                    | <BGP_ROUTE_TYPE_L2VPN_SIGNALING_NLRI>
                    | <BGP_RT_CONSTRAINT>
                    | <PW_ROUTE>

<bgp-route-prefix> ::= <ipv4_route>
                    | <ipv6_route>
                    | <VPN_IPV4_route>
                    | <VPN_IPV6_route>
                    | <Labeled_IPv4>
                    | <Labeled_IPv6>
                    | <FLOW_route>
                    | <EVPN_route> /* draft-ietf-l2vp-evpn */
                    | <MVPN_route> /* RFC6514 */

```



```

| <MCAST_VPLS_route>          /* RFC7117 */
| <MDT_NLRI>                  /* RFC6037 */
| <BGP_LS_route>              /* draft-ietf-idr-ls-distrib
ution */
|
| <VPLS_BGP_NLRI>             /* RFC4761 */
| <L2VPN_SIGNALING_NLRI>     /* RFC6064 */
| <RT_CONSTRAIN>              /* RFC4684 */
| <PW_ROUTE>                  /* RFC7267 */
|
<bgp-attribute-list> ::= [ <bgp_Origin> ]
                        [ <bgp_ASPath> ]
                        [ <bgp_NextHop > ]
                        [ <bgp_MED> ]
                        [ <bgp_LocalPref> ]
                        [ <bgp_ATOMIC_AGGREGATE > ]
                        [ <bgp_AGGREGATOR> ]
                        [ <bgp_CommAttr > ]
                        [ <bgp_ ExtCommAttr > ]
                        [ <bgp_ ClusterList > ]
                        [ <bgp_ORIGINATOR_ID> ]

<ROUTE_DISTINGUISHER> ::= <16-bit ASN: 32-bit-Number >
| <32-bit IPv4_address:16-bit-Number >
| <32-bit ASN:16-bit Number >

<ipv4_route> ::= <IPV4_PREFIX_LENGTH> <IPV4_PREFIX>
<ipv6_route> ::= <IPV6_PREFIX_LENGTH> <IPV6_PREFIX>
<Labeled_IPv4> ::= <LENGTH> <MPLS_LABEL> <IPV4_PREFIX>
<Labeled_IPv6> ::= <LENGTH> <MPLS_LABEL> <IPV6_PREFIX>

<VPN_IPV4>  ::= <LENGTH> <MPLS_LABEL>
                <ROUTE_DISTINGUISHER> <IPV4_PREFIX>
<VPN_IPV6>  ::= <LENGTH> <MPLS_LABEL>
                <ROUTE_DISTINGUISHER> <IPV6_PREFIX>

<FLOW_route > ::= <flow_route_index> <acl>

<EVPN_route> ::= ( <MVPN_ROUTE_TYPE> ( <EVPN_MAC> | <EVPN_AD>
| <EVPN_INCLUSIVE> | <EVPN_ES> )

<EVPN_ROUTE_TYPE> ::= <Ethernet_Auto_Discovery_route>
| <MAC_Advertisement_route>
| <Inclusive_Multicast_route>
| <Ethernet_Segment_route>

<MVPN_route> ::= <MVPN_ROUTE_TYPE> ( <ipv4_prefix> | <ipv6_prefix> )

```

```

<MVPN_ROUTE_TYPE> ::= <Intra_AS_I_PMSI_AD_route>
                    | <Inter_AS_I_PMSI_AD_route >
                    | <S_PMSI_AD_route>
                    | <Leaf_AD_route>
                    | <Source_Active_AD_route>
                    | <Shared_Tree_Join_route>
                    | <Source_Tree_Join_route>

```

```

<MCAST_VPLS_route> ::= <MVPN_ROUTE_TYPE> <Selective_Tree_AD_route>
                    | <Leaf_AD_route>

```

```

/* MDT NLRI - IPv4 address that Identifies the PE
   * that originated this route.
   * The RD identifies the VRF in that PE.
   * The group address MUST be a IPv4 multicast
   *   group address and it is used to build P-tunnel.
   * All PEs attached to given MVPN must specify
   *   the same group address even if the group is
   *   an SSM group. MDT-SAFIs do not carry RTs, and the group
   * address is used to associated a received
   * MDT-SAFI with a VRF.
   */

```

```

<MDT_NLRI> ::= <ROUTE_DISTINGUISHER> <IPV4_ADDRESS> <GROUP_ADDRESS>

```

```

<BGPLS_ROUTE> ::= <BGPLS_ROUTE_TYPE>
                 ( <NODE_NLRI> | <LINK_NLRI>
                   | <IPv4_Topology_Prefix_NLRI>
                   | <IPv6_Topology_Prefix_NLRI>)

```

```

<BGPLS_ROUTE_TYPE> ::= <NODE_NLRI_route>
                    | <LINK_NLRI route>
                    | <IPv4_Topology_Prefix route>
                    | <IPv6_Topology_Prefix route>

```

```

/* VPLS_BGP_NLRI explanation
   * PE participating in the VPLS must have at least one VE ID.
   * VE IDs are typically assigned by the network administrator
   * with a scope local to VPLS. Therefore the VE ID should
   * only belong to one PE unless the CE is multi-homed.
   *
   * if the PE is the VE, then it typical has one VE ID.
   * if the PE is connected to several u-PEs, it has a distinct
   * VE ID for each u-PE. The PE may additionally have a VE ID for
   * itself, if asks acts as a VE for that VPLS.
   * A label block with demultiplexer labels is
   *   used to reach the VE ID.

```

```

* A VPLS BGP NLRI has a VE ID, a VE Block offset (VBO), a
* block offset size (VBS), and a Label base (LB).
* Labels for local VE are (LB+VBS-1) and for the remote
* (VBO+VBS-1).
*/

```

```

<VPLS_BGP_NLRI> ::= <ROUTE_DISTINGUISHER>
    <VE_ID> <VE_BLOCKOFFSET>
    <VE_BLOCKSIZE> <LABEL_BASE>

```

```

<L2VPN_SIGNALING_NLRI> ::= <ROUTE_DISTINGUISHER> <PE_ADDRESS>

```

```

<RT_CONSTRRAIN> ::= <ORIGIN_AS> <ROUTE_TARGET>

```

```

/* LENGTH of PW_route is prefix length of Route Distinguisher +
* Global_ID + Prefix + AC ID in bits
* default PW route = x length
* other: length is between 96-160 bits
*/

```

```

<PW_route> ::= <LENGTH> <ROUTE_DISTINGUISHER>
    <GLOBAL_ID> <PREFIX> <AC_ID>

```

```

<BGP_ROUTE_CREATE> ::= <Not-I2RS_ROUTE> |
    <I2RSCLIENT_CREATE_BGP_ROUTE> |
    <I2RSAGENT_FAILS_BGP_ROUTE_CREATE> |
    <I2RSAGENT_CREATED_BGP_ROUTE> |
    <I2RSAGENT_BGP_ROUTE_CREATE> |
    <I2RSAGENT_REJECTS_BGP_ROUTE_CREATE> |
    <I2RSAGENT_ATTEMPTS_BGP_ROUTE_CREATE> |
    <I2RSAGENT_FAILS_BGP_ROUTE_CREATE> |
    <I2RSAGENT_CREATED_BGP_ROUTE>

```

```

<bgp_state_info> ::= <rib_current_state>
    [ <rib_last_state> ]
    [ <Rib_Rejected_REASON> ]
    [ <Not_Preferred> ]

```

```

<rib_current_state> ::= ( <Active> | <InActive> )
    | ( <primary> | <backup> )
    [ <suppress> ]

```

```

<rib_last_state> ::= ( <Active> | <InActive> )
    ( <primary> | <backup> )
    [ <suppress> ]

```

```

<Rib_Rejected_REASON> ::= <peer_withdraw>
    | <Policy-Results>

```

```

| < Nexthop_Unreachable>
| < Malformed_PEER_DOWN>
| < Malformed_DISCARD>
| < Malformed_IGNORE>

<NOT_Preferred_REASON> ::= <peer_address> | <router_id>
| <Cluster_list_length> | <IGP_METRIC>
| <peer_type> | <origin>
| <weight or preferred_value>
| < Local_Preference | < route_type >
| <AS_PATH_LENGTH> | < MED >

/* BGP Policy */

<BGP-IM_READ_list> ::= [<BGP-IM-Tree-Match ...>]
<BGP-IM_WRITE_list> ::= [<BGP-IM-Tree-Match ...>]
<BGP-IM-Tree-Match> ::= <BGP-IM-Tree-Match-protocol-instance>
<BGP-IM-Match-Protocol-instance> ::= (<BGP_protocol> ...)

/* Policy Rule Extensions for BGP */
<Policy-Rule_Match_Nodes_BGP-IM> ::=
    (<Policy-Rule_Match_node_BGP-IM> ...)
<Policy-Rule_Match_Values_BGP-IM> ::=
    (<Policy-Rule_Match_value_BGP-IM> ...)
<Policy-Rule_Match_operators_BGP-IM> ::=
    (<Policy-Rule_operators_BGP-IM> ...)

<Policy-Rule_match_node_BGP-IM> ::= [<Policy-Rule_Match_acl> ]
    [<as-path-filter>]
    [<community-filter>]
    [<MED>]
    [<extcommunity-filter>]
    [<interface>]
    [<ipv4-route-list>]
    [<ipv6-route-list>]
    [<peer-address-list>]
    [<MPLS_LABEL>]
    [<rd-filter>]
    [<TAG>]

<Policy-Rule_Action_Vars_BGP-IM> ::= [<adc_path_list>]
    [<adc_comm_list>]
    [comm_mod_args]
<acl> ::= <Policy-Rule_match

/* BGP policy_lists */
<peer-address-list> ::= (Peer_Session_Address ...);

```

```

<ipv4-route-list> ::= (<ipv4_route> ..)
<ipv6-route-list> ::= (<ipv6_route> ..)

<BGP_adc_path_list>:: =      [<add_as_path_list> |
                             <delete_as_path_list>
                             <modify_as_path_list>]

<adc_comm_list>::=  [<add_community_list> |      /* community list */
                    [<delete_community_list>|
                     <modify_community_list>
                     <modify_med>]

<comm_mod_args>::=  [<modify_next_hop >]
                    [<modify_local_preference>]
                    [<modify_mpls_label>]
                    [<modify_origin>]
                    [<modify_ip_dscp>]
                    [<modify_preference>]
                    [<modify_bgp_preference_value>]
                    [<modify_bgp_preference_weight>]

```

## 6. IANA Considerations

This draft includes no request to IANA.

## 7. Security Considerations

TBD.

## 8. Informative References

[I-D.hares-i2rs-info-model-policy]

Hares, S. and W. Wu, "An Information Model for Network policy", draft-hares-i2rs-info-model-policy-02 (work in progress), March 2014.

[I-D.ietf-i2rs-architecture]

Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-04 (work in progress), June 2014.

[I-D.ietf-i2rs-rib-info-model]

Bahadur, N., Folkes, R., Kini, S., and J. Medved, "Routing Information Base Info Model", draft-ietf-i2rs-rib-info-model-03 (work in progress), May 2014.

- [I-D.ietf-netconf-restconf]  
Bierman, A., Bjorklund, M., Watsen, K., and R. Fernando,  
"RESTCONF Protocol", draft-ietf-netconf-restconf-00 (work  
in progress), March 2014.
- [I-D.keyupate-i2rs-bgp-usecases]  
Patel, K., Fernando, R., Gredler, H., Amante, S., White,  
R., and S. Hares, "Use Cases for an Interface to BGP  
Protocol", draft-keyupate-i2rs-bgp-usecases-03 (work in  
progress), June 2014.
- [I-D.white-i2rs-use-case]  
White, R., Hares, S., and A. Retana, "Protocol Independent  
Use Cases for an Interface to the Routing System", draft-  
white-i2rs-use-case-05 (work in progress), June 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3060] Moore, B., Ellesson, E., Strassner, J., and A. Westerinen,  
"Policy Core Information Model -- Version 1  
Specification", RFC 3060, February 2001.
- [RFC3644] Snir, Y., Ramberg, Y., Strassner, J., Cohen, R., and B.  
Moore, "Policy Quality of Service (QoS) Information  
Model", RFC 3644, November 2003.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash,  
"Policy-Enabled Path Computation Framework", RFC 5394,  
December 2008.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax  
Used to Form Encoding Rules in Various Routing Protocol  
Specifications", RFC 5511, April 2009.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration  
Protocol (NETCONF) Access Control Model", RFC 6536, March  
2012.

#### Authors' Addresses

Susan Hares  
Huawei  
7453 Hickory Hill  
Saline, MI 48176  
USA

Email: shares@ndzh.com

Lixing Wang  
Huawei  
Beijing  
PRC

Email: wanglixing@huawei.com

Shunwan Zhuang  
Huawei  
Beijing  
PRC

Email: Zhuangshunwan@huawei.com