I2RS Working Group L. Dunbar
Internet-Draft S. Hares
Intended status: Informational Huawei
Expires: September 25, 2015 J. Tantsura
Ericsson
March 24, 2015

An Information Model for Filter Rules for Discovery and Traffic for I2RS Filter-Based RIB
draft-dunbar-i2rs-discover-traffic-rules-00

Abstract

This draft describes an I2RS Filter RIB information model for managing routers to steer traffic to their designated service functions or service function instances via the I2RS interface. The purpose of these filters is to guide the specific flows traversing their assigned Service Function Chains in the network.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This draft describes an I2RS Filter RIB information model for managing routers to steer traffic to their designated service functions or service function instances via the I2RS interface. The purpose of these filters is to guide the specific flows traversing along their assigned Service Function Chains in the network.

The I2RS Filter-Based RIB (FB-RIB) is described in [I-D.kini-i2rs-fb-fib-info-model]. I2RS FB-RIBs are protocol independent RIBs.

An I2RS Filter-Based RIB (FB-RIB) is an entity that contains an ordered set of filters (match/action conditions) and a default RIB of the form found in [I-D.ietf-i2rs-rib-info-model] An ordered set of filters implies that the insertion of a filter router into a FB-RIB must allow for the insertion of a filter-route at a specific position and the deletion of a filter at a specific position. The ability to change a route combines these two functions (deleting existing filter route rule and adding a new policy route). Each I2RS FB-RIB is contained within a routing instance, but one routing instance can contain multiple FB-RIBs. Each routing instance is associated with a set of interface, a router-id, a default RIB.

[I-D.kini-i2rs-fb-fib-info-model] describes a generic filter form which has specific filters for L1, L2, L3, and Service level RIBs. This document describes the FB-RIB filters for the following types of service level data forwarding:

o a) Traffic flow steering rules on a router for specific Service, Function Path (SFP) or Rendered Service Path (RSP).

o b) service function instance discovery traffic (E.g. ARP, ND, or other broadcast/multicast data).

I2RS dynamic interface augments the service function configuration, status, and OAM information. This augments yang data models proposed in [I-D.penno-sfc-yang] and [I-D.xia-sfc-yang-oam]. These SFC yang module documents have not been adopted by the SFC WG, but the best indication of this work.

Section 3 of this document provides Service-chaining related background for this Information model. This includes background on service function chaining, deployment of service chaining, requirements for I2RS in service chaining.

Section 4 provides background on the generic I2rs Filter-Based RIBS, an how these service level traffic filters fit into that generic model.

Section 5 contains the description Information Model and Yang data model for traffic flow steering rules.

Section 6 contains the description of the Information Model for service function instance discovery traffic and Yang data model for service function instance filters.

Section 7 contains the description of the I2RS SFC yang components the traffic features depend on. These service features are being worked on by the SFC WG so shared definitions are necessary.

Section 8 contains the security considerations for use of a data model that may arise from this information model. This Information Model is only an intermediate step on the pathway to a deployable yang data model.

2. Terminology

FB-RIB: Filter-Based Routing Information Base

The I2RS protocol independent RIBs operate on a set of interfaces, and contain a ordered list of filter rules (match-condition rules).

NFV: Network Function Virtualization

[NFV-Terminology].

RSP: Rendered SErvice Function Path (RSP)

[I-D.ietf-sfc-architecture]

Service Chain

[I-D.bitar-i2rs-service-chaining] defines a service chain as an ordered set of services applied to a packet of flow. An example of this is a sequence of service function such as Chain#1 {s1, s4, s6} or Chain#2{s4, s7} at functional level. Also see the definition of Service Function Chain in [I-D.bitar-i2rs-service-chaining]

Service Chain Instance Path

The actual Service Function Instance Components selected for a service chain.

SF: Service Function

[I-D.ietf-sfc-problem-statement].

SFF: Service Function Forwarder

SFFN: Service Function Forwarder Node

[I-D.bitar-i2rs-service-chaining]states service function can run: a) natively within a router (or routing system), b) on a virtual machine on a server or service engine, or in a dedicated standalone hardware appliance.

SFFaddr: Service Node Address

[I-D.ietf-sfc-problem-statement] states this address should be IP Address, or tuple of (SFFaddr, host system IP address) or tuple of (host system IP address, system internal ID for service engine).

Service Type

[I-D.ietf-sfc-problem-statement].

VNF: Virtualized Network Function

[NFV-Terminology]

Virtual Network Instance Identifier

Virtual Network Instance ID

3. Informational Model Background- SFC

Section 3.1 provides the background on service function chaining (SFC), and section 3.2 provides the I2RS use case requirements for the basic service chaining. Section 3.3 provides the overview of how filter rules for traffic flow for specific service function paths (SFPs) and rendered service paths (RSPs). Section 3.4 provides the background on service function instance discovery traffic and how the need for traffic filters.

Sections 3.5 provides information on SFC-USE-REQ01 from [I-D.ietf-i2rs-usecase-reqs-summary] which specifies requirements related to the filtering of service chaining traffic flows.

Section 3.6 provides information on SFC-USE-REQ02 use case from the same document. SFC-USE-REQ02 is related to handling service-discovery traffic flows.

Section 3.7 describes Section 3.7 describes the following I2RS use case requirements: SFC-Use-REQ03, SFC-USE-REQ04, SFC-USE-REQ05, and SFC-USE-REQ06. These use case requirements define SF and SFF information which may be necessary for the I2RS Client to process data related to the SFF traffic filters or service discovery traffic.

3.1. Service Function Chaining

The Service Function Chain (SFC) [I-D.ietf-sfc-architecture] is defined as an ordered set of abstract service functions (SFs) that must be applied to packets and/or flows that meet certain criteria.

The criteria of assigning packets to a service function chain can vary, some can be based on L3/L2 header fields, some can be based on L4 header, and some can be based on L5-L7 header, packet size, special events, or combination of all above. A match filter can be created either by long-term configuration or by the I2RS dynamic interface.

For Service Chain with matching criteria that are beyond L2/L3 header, such as L4-L7 header or other events, it is more economical to have some specialized nodes with DPI capability to inspect the packets and associate an identifier in the L2/L3 header to the packets that match the SFC criteria. By doing so, the subsequent routers/switches only need to forward based on the identifier (a.k.a. Service Chain identifier). Again, Filters that examine service chain identifiers prior to forwarding traffic can be configured or dynamically created in the I2RS FB-RIB.

```
                                    |1  -----   |n        |21   ---- |2m
                  +---+---+   +---+---+   +-+---+   +--+-----+
                  | SF#1  |   |SF#n   |   |SF#i1|   |SF#im   |
                  |       |   |       |   |     |   |        |
                  +---+---+   +---+---+   +--+--+   +--+--+--+
                      :           :          :         :  :
                      :           :          :         :  :
                       \         /            \       /
     +--------------+   +--------+             +---------+
 -- >| Chain        |   | SFF    |   ------    | SFF     | ---->
     |classifier    |   |Node-1  |             | Node-i  |
     +--------------+   +----+---+             +----+--+-+
                   \         |                     /
                    \        | SFC Encapsulation  /
                     \       |                   /
 ,. ......................................._
 ,-'                                        `-.
 /                                              `.
 |                      Network                   |
 `.                                             /
 `.__.................................. _,-'
```

Figure 1   Framework of Service Chain

IETF SFC WG has been chartered to create a new Service Chain Header that can carry Service Chain ID plus metadata or/and the actual service function path in the header. However, not every service chain implementation requires the newly created service chain header. BESS WG is working on service function chains using existing MPLS/BGP as the tunnel and/or chain control.

[I-D.boucadair-sfc-design-analysis] describes several Service Function Chain mechanisms that do not use new Service Chain Header.

This draft describes an I2RS information model for

managing Chain Classifier node to assign specific identifier to the packets that match specific criteria via the I2RS interface,

managing routers to steer traffic to their designated service functions or service function instances via the I2RS interface, and

retrieving SF connectivity to SFF via the I2RS interface for Topology Discovery.

A service chain path identifies the exact SFF nodes and SF sequence visited by each SFF node for a specific service function chain.

3.2. Installing Service Function Chain steering filters using I2RS

It is assumed that there is an external service function chain manager or an orchestration system that computes the Service Function Path including the sequence of SFF nodes and the sequence of service functions for flows to traverse within each SFF node. A service chain path identifies the exact SFF nodes and SF sequence visited by each SFF node for a specific service function chain.

It is beyond the scope of I2RS and this draft on how the Service Function Chain orchestration system computes the path.

This Service Chain Orchestration System behaves as an I2RS client and uses I2RS interface to instruct routers what filter rules to dynamically install to guide traffic to and along service chain paths as shown in figure 2. The I2RS filter rules include filter classification rules (match rules) and action upon matches forwarding rules, encapsulation rules to next-hop service function, or next-hop SFF nodes).

The SFF Shim in the diagram below groups the additional work needed to for Service Functions and pass the steering policies to FB-RIB Manager described in [I-D.kini-i2rs-fb-fib-info-model]. Here is the extra work needed by SFF agent:

- Managing the mapping between Service Function Chain identifier (SFC-identifier) and the local identifier on the link to service functions. Some service functions do not terminate the Service Chain ID carried by the packets; some service functions need a different identifier, such as VLAN to differentiate flows.

- Managing reachability to directly attached service functions,

- Managing balancing among multiple ports that connected to different instances of the same service function type.

The SFF Shim can be implemented as part of the orchestrator or as part of an I2RS broker. This document focuses on the I2RS Client-1 to I2RS Agent-2 communication which may need to query or modify the above functions.

```
            +------------------------------------------------+
            |Service Function Chain Manager or Orchestration |
                    |     Shim - SFF                                 |
                    |                                                |
            |              I2RS client 2                     |
            +-----------------+------------------------------+
                              |
                              V
                   +----------+----------+
                   |     I2RS Agent 1    |
                   | +-------+ +-----+   |
                   | |FB-RIB | | RIB |   |
                   | +-------+ +-----+   |
                   +---------------------+
                   |    Routing System   |
                   +---------------------+
                              ^
                              |
             +---------------------------------+
             |                                 |
             V                                 V
      +-------------+                   +-------------+
      |FIB manager 1|                   |FIB manager M|
      |   +-----+   |    ..........     |   +-----+   |
      |   | FIB |   |                   |   | FIB |   |
      |   +-----+   |                   |   +-----+   |
      +-------------+                   +-------------+
```

Figure 2   SFF Shim Layer in relation to RIB Manager

The SFF client must be able to instruct the I2RS Agent to

- Add/Modify/Delete the filter routes in the FB-RIB based on SFF reachability and SF reachability (locally attached Service functions),

- Add/Modify/Delete filter routes in the FB-FIB that direct load balancing for SFF reachability or SF reachability,

- Allow FB-RIB filter routes that match a service function identifier to have a forwarding action via interfaces, local-links, tunnels or L3 nexthops or Service layer next-hops. (These type of features are utilized in the I2RS RIB Model ([I-D.ietf-i2rs-rib-info-model] and [I-D.wang-i2rs-rib-data-model]).

3.3. SFC Service Layer Steering Policies

The SFF nodes are interconnected by tunnels (e.g. GRE or VxLAN) and the SF are attached to a SFF node via Ethernet link or other link types. Therefore, the steering policies to a SFF node for service function chain depends on if the packet comes from previous SFF or comes from a specific SF. Due to this fact, the SFC Service Layer Steering filter routes need to be able to specify the ingress port/ interface in the filter match.

There are multiple different steering policies for one flow within one SFF and each set of steering policies is specific for an ingress port/interface.

```
                  figure 3

              Ingress Port match
                           |
                           |
       +-------+--------+--+------+-------+-------+-------+-------+
       |       |        |         |       |       |       |       |
       |       |        |         |       |       |       |       |
     L3Header L2header L4 header VLAN    VN ID   size    event ..
```

The action has to be egress port specific.

3.4. Service Function Instances Discovery

Service Function Instance Discovery is not required to have Service chain forwarding, but this function may provide a useful service in many networks.

A Service function instance can be either attached to a router via a physical interface or instantiated on a virtual machine that is attached to a router. However, not every attached host to a router is a service functions.

It is assumed that the Service Function Chain Manager or Orchestration can get all the IP addresses or IP prefix of the service function instances from an external authoritative entity, such as a database or provisioning system. However, the SFC orchestration may not know how/where the service function instances are attached to the network, especially in an environment where virtualized hosts can be moved easily.

Here is the procedure for Service Chain Orchestration system to discover how/where service function instances are attached in the network:

1) The Service Chain Manager or orchestration can passed the Service function addresses or prefix to the relevant SFFs. The SFFs can send ARP/ND broadcast/multicast messages to all the attached nodes.

2) Service function instances will respond to ARP (IPv4)/ND (IPv6) requests from its L2/L3 boundary router.

3) SFF nodes can report the directly reachable Service function instances to the Service Chain Manager/Controller.

```
                      Service Chain Manager/Controller
                               ^   |
                               |   | A: Set filter for
            B:                 |   |    the interested service
            Router reports the |   |    function instances
         Directly attached     |   |
         Service Function      |   |
         Instances             |   V
                        +------+---+-------------+
                        |     Router             |
                        ++-----+----------+------+
                        /      |          |       \
                       /       |          |        \
                    +-+-+    +-+-+      +-+-+     +-+-+
                    |   |... |   |      |   | ... |   |
                    +---+    +---+      +---+     +---+ Server racks
                    |   |... |   |      |   | ... |   | for hosting
                    +---+    +---+      +---+     +---+ Service
                    |   |... |   |      |   | ... |   | Function
                    +---+    +---+      +---+     +---+ Instances
```

Figure 1: Service Function Instances

3.5. I2RS Use Case Requirements for Service Flow Filtering

This section reviews the requirements for Flow Filtering Policies for SFFNs within the SFC domain.

Inherent in the [I-D.ietf-sfc-problem-statement] is the need for policies that establish and filter data flows on the SFFs along the Service Function Chain path. The SFC use case

[I-D.bitar-i2rs-service-chaining] and the [I-D.ietf-i2rs-usecase-reqs-summary] suggest the SFF resources that must be on each SFF Node (SFFN). The SFFN resources include the following elements that the I2RS Client-I2RS Agent protocol can utilize in filters:

SFC-Use-REQ01:Address (R)

has the following address requirements:

* IP address

* service-node tuple (service node IP address, Host system address)

* host-node tuple (hosting system IP-address, system internal identifier)

3.6. I2RS Use Case Requirements Related to Service Discovery Traffic

The following I2RS Use Case Requirement specifies the following additional information which may be used by the SFF SHIM layer (figure 2)

SFC-Use-REQ02:Supported Service Types (R/W)

abstract service function type, or can be vendor specific service function types.

Note: The current SFC WG suggest hat the SFF does not need to know the SF type on the node in order to steer the data to their designated service function. However, the information can help is the service discovery.

3.7. I2RS Use Case Requirements Related to SFF SHIM function

The I2RS Use Case Requirements specify the following additional information that this draft suggest may be used by the SFF SHIM layer (figure 2) to calculate flow filters. These features are the following:

SFC-Use-REQ03:Virtual contexts (R/W)SHOULD include:

* Maximum Number of virtual contexts supported

* Current number of virtual contexts in use

- Number of virtual contexts available
- Supported Context (VRF)

SFC-Use-REQ04: Customers currently on node (R)

SFC-Use-REQ05: Customer Support Table (per customer ID) (R)

with the following contents per entry:

- Customer-id
- List of supported Virtual Contexts

SFC-Use-REQ06: Service Resource Table (R/W)

which includes:

- index: Comprised of service node, virtual context, service type
- service bandwidth capacity
- supported packet rate (packets/second)
- supported bandwidth (kps)
- IP Forwarding support: specified as routing-instance(s), RIBs, Address-families supported
- Maximum RIB-size
- Maximum Forward Data Base size
- Maximum Number of 64 bit statistics counters for policy accounting
- Maximum number of supported flows for services

SFC-Use-REQ07: Virtual Network Topology (VNT) (R)

which includes:

- topology of access points

4. Filter-Based RIB Background

Filter based (FB) routing matches fields in the IP header plus other higher layer packet information. Filters with a match-action pair allow the filters to impact the forwarding of packets. Actions may impact forwarding or set something in the packet that will impact forwarding.

A Filter-Based RIB (Routing Information Base) contains a list of filters (match-action conditions) and a default RIB of the form found in [I-D.ietf-i2rs-rib-info-model] The default RIB routes any packet not matched by the order list of filter rules. An order set of filters implies that the I2RS agent must be able to insert a filter route at a specific position and delete a filter route at a specific position. Changing a route is simply a combination of the two (delete a route and add a new route).

The Filter-Based RIB found in [I-D.kini-i2rs-fb-fib-info-model] allows for a generic filter that supports L1, L2, L3, and Service matches in the match-condition, or a more specific match-condition filter (E.g. ACL filters found in [I-D.ietf-netmod-acl-model].

Each Filter-Based RIB (FB-RIB)is contained within a routing instance, but one routing instance may contain multiple RB-FIBs. In I2RS Models, each routing instance is associated with a set of interfaces, a router-id, a list of I2RS RIBs, and a list of FB-RIBs. Only some of the interfaces within the routing instance may be associated with a FB-RIB. Each FB-RIB also designates a default destination-based RIB (FB-RIB Default RIB) that forward traffic not matched by the filters. Any traffic not match by the FB-RIB filters or the FB-RIB Default RIB is dropped.

Packets arriving on an interface associated with an FB-RIB will be forwarded based on a match to the filters in a FB-RIB associated with that interface. The processing of the packet does the following:

- if a packet successfully matches, the rule-actions are applied.
- If a packet does not successful match a filter, the filter route processing goes to the next filter in the list. This continues until all filter routes are matched.
- If no match has been found within the FB-RIBs on the FB-RIB list, then the packet will be forward to the FB-RIB Default RIB specified by the FB-RIB. If non-exists, then the packet will be discarded.

- If no match is found in the FB-RIB Default RIB, the packet will be discarded.

5. Information Model for Traffic steering rules

The semantics of traffic steering rules is "Match" and "Action". This draft uses the generic match-action filters described in [I-D.kini-i2rs-fb-fib-info-model] which provides filters at L1, L2, L3, L4, Service packets

The match filters for SFF need to support the fields in a packet and packet meta-data:

- Layer 2: ingress port, destination MAC, source MAC, VLAN ID, GRE Keys, and L2 packet size;

- Layer 3:MPLS label, destination IP, source IP, VN-ID, layer 3 packet size,

- Layer 4: TCP port and UDP port,

- Service Chain Identifier (Service-level)

The generic match-action filters provide a generic filter format for match actions for packets that examine these L1-L4, and service layer fields.

A SFF node may not support some of the matching criteria listed above. It is important that Service Function Chain Orchestration System can retrieve the type of FB-RIB filters supported matching criteria by I2RS agent in the SFF nodes.

The Actions for traffic steering could be to steer traffic to the attached service function via a specific port with specific VLAN-ID added, or forward traffic to the next SFF node(s) with specific VxLAN header.

When steering to the attached service function, the action may include such things as:

- adding VLAN-ID tags,

- removing service header fields of a packet have to be removed if packets with a certain header are not supported by the attached service functions;

- Forwarding traffic out a particular interface or tunnel.

5.1. 5.1 Existing FB-RIB information in RBNF Form

```
     <FB-RIB-match-action>::= [<BNP_GENERIC-MATCH_ACTION>
                                 <generic-match-action-rule>] |
                                [<ACL_MATCH_ACTION><acl-list-entry-name>]

     <generic-match-action-rule>::= <bnp-term-match><bnp-action>
                                    <bnp-forward>

      <bnp-term-match>:: = <interface-match>
                           <L1-match>
                           <L2-match>
                           <L3-match>
                           <L4-match>
                           <Service-header-match>

         <bnp-action>::= <NACTIONS>
                     <qos-actions>
                     <forwarding-actions>

        <qos-actions>::= <L1-qos-action>
                     <L2-qos-action>
                     <L3-qos-action>
                     <L4-qos-action>
                     <Service-qos-action>

    <bnp-forward>:: <fb-std-forward> <fb-std-drop>
    <fb-std-forward>::= [<INTEFACE-ID> [<preference>]] |
                        [<rib-nexthop><rib-attributes>]

     <fb-std-drop>::= <Forward><Drop>

        # Generic interface filter from RIB and FB-FIB
    # Assumed from generic filtering yang document

          <interface-match>:: = <interface-list> <port-list>
      <interface-list>:: = [<INTERFACE_IDENTIFIER> . . . ]
      <port-list>::==  [<PORT_IDENTIFER> . . . ]

      <L2-match>:: = [<L2-type-match_entry> . . .]
      <l2-matches_entry>]::=[<L2-DMAC-MATCH> <destination-mac>]
               [<L2_SMAC-MATCH> <source-mac>]
               [<L2_DMAC-SRC-MATCH><destination-mac><source-mac>]
               [<L2_MMAC-DMAC-MATCH> <multicast-mac>]
               [<L2_VLAN-Match> <vlan-id>]
                   [<L2-packet-size>]
```

```
                  <L3-match>::= [<L3-match-entry> ...]

                  <L3-match-entry>::= [<LABEL_MATCH> <label>]
                           [<DESTINATION_ADDRESS_MATCH> <ip-address>]
               [<SOURCE_ADDRESS_MATCH><ip-address>]
               [<DESTINATION-SOURCE_ADDRESS_MATCH>
                    <ip-address><ip-address>]
                [<IP-Packet-Type><ip-packet-type>]
                [<IPv6-Flow-Type><ipv6-flow-id>]
                [<L3-packet-size>]


        <ip-address> ::= <IPV4_ADDRESS>
                    |<ipv4-prefix >
                    |<IPV6_ADDRESS>
                    |<ipv6-prefix >]

                 <ip-packet-type>::=[<IPv4><IPv4-packet-type>]
                               [<IPv6><IPv6-packet-type>]
 # this is a partial list of the all types
 # which is used by this IM model
#
        <IPv4-packet-type>:: = <ARP><ICMP>
        <IPv6-packet-type>:: == <ND>
        <L4-match>:: [<TCP_PORT> | <UPD-Port> ]

    <label> ::= [<MPLS_LABEL>] | <GRE-KEY>
     <L4-field>::= <TCP_PORT> | <UDP-PORT>

        <Service-Match>::== [<SERVICE_CHAIN_MATCH><service-chain-matches>]
                [<L3VPN_SERIVCE_MATCH><L3VPN-feature-matches>]

        <Service-QOS-Actions::> == [<SERVICE_CHAIN_QOS>
                 <service-chain-qos-actions>]
                                  [<L3VPN_QOS_ACTIONS>
                                          <l3vpn-qos-actions>]
```

5.2. 5.2. SFF Filters in RBNF Form

```
#definitions unique to the SFF filter rules
# SFF

  <Service-chain-matches>::= <SF-MATCH-NAME><sf-filter-name>
         [<SF-MATCH><sf-match> . . .  ]
                  [<SFF-MATCH> <sff-match> . . . ]
                  [<SF-MATCH><sf-match> . . . ]
                  [<SFC-ID-Match> <sfc-match> . . . ]
                  [<SFC-Client-Match><service-client-identifier> . . . ]
```

```
<sf-filter-name> = <SF_FILTER_NAME>

#Section 7 povides the definition for <service-client-identifier>


# Service function definition comes from the [I-D.penno-sfc-yang-13]
# (Note: Additional filters may be added here]

        <sf-match>:: = [<SF_TYPE><sfc-sft-service-function-type>]
                        [<SF_NAME><string>]
                        [<SF_REST-URI><inet:uri>]

                <sfc-match>::= <SFC_CHAIN_NAME><string>

# Service Chain QOS functions

  <service-chain-qos-actions>::=[<sff-qos-action>]
                                                                    [<isfi-q
os-actions>]


        <sff-qos-action>::= [<SFF-ingress-action>]
                                                    [<SFF-steering-action>]
                        [<SFF-egress-action>]

        <SFF-ingress-action> :: = [<sff-ingress-vlan>]
                                            [<sff-ingress-mac>]

     <sff-ingress-vlan> :: = [<VXLAN_REMOVAL_TYPE]
                        <decapsulate-VxLAN-header>
                                                  <filter-decapsulated-packet>

     <sff-ingress-mac>::= [<MAC_HEADER_REMOVAL_TYPE>
                                                           <remove MAC-Header>
                                                          <encapsulate-ID>]

    <SFF-egress-action> :: <encapsulation-metho0d>
                                                             | <metadata>

    <metadata> ::= [<ATTACH> <object>] |
                        <detach>

     <encapsulation-method>::=<add-VXLAN-header>
                             |<add-VLAN>
```

6. 6. Information Model for Interested Service Function Instances

Service Function Instances placement can be managed by entities that are not integrated with Service Chain Manager. Therefore, it is necessary for the Service Chain Manager to discover all the Service Function Instances that might be needed for a specific service chain. Service Chain Manager can send down the filter periodically or on-demand (i.e. when there is a request for building a specific service chain for a client).

Some service function instances are attached to router via tunnels, e.g. VxLAN. Service Function Instances might be partitioned by clients, which are differentiated by different network ID (e.g. VNID, VPN ID, etc). Some filter will carry the network ID (tenant ID, or VPN ID) to get specific service functions.

The service chain manager/controller acts like an I2RS Client to communicate with the I2RS Agents operating in the router or I2RS Agents operating on the service function instances in the server racks to discover and control specific service function instances.

The I2RS Client-Agent must be able to discover the I2RS Agent associated with a specific Service Function instance by querying for: SFFN Address, SFFN type, or SFFN virtual context or SFFN Customer.

```
          <issf-match-match-filters>::= <sfc-filter-name>
                  [ <SFC-Client-Match><service-client-identifier>
           [[ <L3-match-entry>] ...  ]


        <client-identifier> ::= <client-identifier-type>
                                <client-identifier >
        <client-identifier-type> ::= <GRE>
                               | <VxLAN>
                                | <NVGRE>

        <client-identifier > ::= (<VXLAN> <VXLAN_IDENTIFIER>)
                                | (<NVGRE> <VIRTUAL_SUBNET_ID>)
                                | (<GRE> <GRE_KEY>)
```

6.1. RPC Information Model for Reporting Directly Attached Instances

When a router receives the filter of the interested Service Function Instances, it can scan through all its interfaces to check if any of the addresses in the filter list are attached to the interfaces. For the Service Function Instances attached via Layer 2, the router can send ARP/ND to get the matching instances to respond. For the

Service Function Instances attached via Layer 3, the router can use Ping to check if the addresses in the filter are attached.

```
 #This RPC assumes FB-RIB filter is there but inactive
 #
    rpcs:
     +--x-Check-Attached-IP-Address-in-Filter
           +--FB-RIB-Rule FB-RIB:rule:rule-name
           +--FB-RIB-rule-group FB-RIB:rule-group


 # The response should be grouped by SF-FILTER-NAME per routing instance
 #response should be
  +--x-response-
     +--ro instance-name
     +--ro FB-filter-name
     +  ro sfc-filter-name
     +--ro attached-address
        +--ro IPv4-address-list
        +--ro IPv6-address-list
```

6.2. RBNF for Reporting Directly Attached Instances

RBNF for Reporting Directly Attached Instances

```
          <sf-instance-list> ::= <INSTANCE-LIST-NAME>
                      < SF-FILTER-NAME >
                      [<INTERFACE_IDENTIFIER>
                      |<ipv4-address-list>
                      |<ipv6-address-list>]]
```

7. Service Function Forwarder Nodes I2RS Information

The following I2RS constructs are necessary to support the service function forwarder node functions required. These functions may be needed by the SFF shim material. These functions are not contained in [I-D.penno-sfc-yang] or [I-D.xia-sfc-yang-oam]. If the SFF_node related information structures are global configuration/state functions, then these should be added to SFC to [I-D.penno-sfc-yang] and I2RS definitions can share grouping definitions with this I2RS state.

```
        <SFF_node> ::= <SFFN_address>   /*SFC-Use-REQ01 */
                [<Attached_Service_node>]       /*SFC-Use-REQ02 */
```

```
               [<SFFN_virtual_contexts>]       /*SFC-Use-REQ03 */
               [<SFFN_customer_cnt>]    /*SFC-Use-REQ04 */
               [<SFFN_Customer_support_table>] /*SFC-Use-REQ05 */
               [<SFFN_Service_Resource_table>] /*SFC-Use-REQ06 */
               [<SFFN_VNTopo>]                 /*SFC-Use-07*/

               <SFFN_address> ::== <ip_address>

               <Attached_Service_node> ::=
                               | [ (<service-node-ip_address>
                                       <host-system-ip_address>)]
                               | [ (<hosting-system-ip_address>
                                   <system-internal_ID>)]

               <service-node-ip_address> ::= <ip_address>
               <host-system-ip_address> ::= <ip_address>
               <hosting-system-ip_address> ::= <ip_address>
               <system-internal_ID> ::= INTEGER-64;

               /* SFC-Use-02 */
               <SFFN_supported_types> ::= <Attached_Service_node_types>


               /* These are the types specified by the SFC-REQ-02]
               <SF_Types> ::= [<SF_TYPE_FW>]
                                                                   [<SF_TYPE
_LB>]
                                                                   [<SF_TYPE
_DPI>]
                                                                   [<SF_TYPE
_NAT>]
               /* SFC-Use-03 */                                         ...
               <SFFN_virtual_contexts> ::== <VContext_max>
                                       <VContext_current_inuse>
                                       <VContext_current_avail>
                                       <SFFN_Types>

               /*SFC-Use-04 */
               <SFFN_customer_cur_cnt> ::= INTEGER;

       /* SFC-Use-05: Customer Support Table per Customer ID */
               <SFFN_customer_table> ::= [<SFFN_customer< ...]

               <SFFN_customer> ::= <SFFN_customer_Name>
                                                 <SFFN_customer_ID>
                                                 <SFFN_customers_contexts>

               <SFFN_customers_contexts> ::= <SFFN_Types>

               /*SFC-Use-REQ06 */
               <SFFN_Service_Resource_table> ::=  <SFF_Service_resource_ind
ex>
```

```
                                        <SFFN-SR_service_BW_capacity>
                                        <SFFN-SR_packet_rate_max>
                                        <SFFN-SR_BW>
                                        <SFFN-SR_IP_fwd_instance_list>
                                        <SFFN-SR_MAX_RIB>
                                        <SFFN-SR_MAX_FIB>
                                        <SFFN-SR_MAX_COUNTER64>
                                        <SFFN-SR_MAX_Flows>

                        <SFF_Service_resource_index> := <SFFN_Address>
                                        <VContext_ID>
                                        <Service_types>

                /*SFC-Use-REQ07
                 * SFC_topology is defined by
                 * ietf-hares-i2rs-service-topology
                 * which includes node code
                 */
                <SFF_VNT> ::= <SFC_Topology>
```

8. Security Considerations

The SC use cases described in this document assumes use of I2RS programmatic interfaces described in the I2RS framework mentioned in [I-D.ietf-i2rs-architecture]. This document does not change the underlying security issues inherent in the existing in [I-D.ietf-i2rs-architecture].

I2RS FB-FIBs will filter packets in the traffic stream, modify packets (via actions), and forward data packet. These I2RS FB-RIB filters operate dynamically on the the packets. The FB-RIB filters in the I2RS Agent may in turn be changed dynamically by the I2RS Client. The dynamic nature of the changes does not change the fundamental actions of routers, but rate is change is increased.

9. IANA Considerations

This draft includes no request to IANA.

10. Acknowledgements

We’d like to thank Qin Wu for his comments on this document relating to the service topologies.

11. References

11.1. Normative References

[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-09 (work in progress), March 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[I-D.bitar-i2rs-service-chaining]
Bitar, N., Heron, G., Fang, L., ramki, r., Leymann, N., Shah, H., and W. Haddad, "Interface to the Routing System (I2RS) for Service Chaining: Use Cases and Requirements", draft-bitar-i2rs-service-chaining-01 (work in progress), February 2014.

[I-D.boucadair-sfc-design-analysis]
Boucadair, M., Jacquenet, C., Parker, R., and L. Dunbar, "Service Function Chaining: Design Considerations, Analysis & Recommendations", draft-boucadair-sfc-design-analysis-03 (work in progress), October 2014.

[I-D.hares-i2rs-info-model-policy]
Hares, S. and W. Wu, "An Information Model for Basic Network Policy", draft-hares-i2rs-info-model-policy-03 (work in progress), July 2014.

[I-D.hares-i2rs-info-model-service-topo]
Hares, S., Wu, W., Wang, Z., and J. You, "An Information model for service topology", draft-hares-i2rs-info-model-service-topo-03 (work in progress), January 2015.

[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-06 (work in progress), March 2015.

[I-D.ietf-i2rs-usecase-reqs-summary]
Hares, S. and M. Chen, "Summary of I2RS Use Case Requirements", draft-ietf-i2rs-usecase-reqs-summary-00 (work in progress), November 2014.

[I-D.ietf-netmod-acl-model]
Bogdanovic, D., Sreenivasa, K., Huang, L., and D. Blair, "Network Access Control List (ACL) YANG Data Model", draft-ietf-netmod-acl-model-02 (work in progress), March 2015.

[I-D.ietf-sfc-architecture]
Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-07 (work in progress), March 2015.

[I-D.ietf-sfc-problem-statement]
Quinn, P. and T. Nadeau, "Service Function Chaining Problem Statement", draft-ietf-sfc-problem-statement-13 (work in progress), February 2015.

[I-D.kini-i2rs-fb-fib-info-model]
Kini, S., Hares, S., Ghanwani, A., Krishnan, R., Wu, Q., Bogdanovic, D., Tantsura, J., and R. White, "Filter-Based RIB Information Model", draft-kini-i2rs-fb-fib-info-model-00 (work in progress), March 2015.

[I-D.medved-i2rs-topology-requirements]
Medved, J., Previdi, S., Gredler, H., Nadeau, T., and S. Amante, "Topology API Requirements", draft-medved-i2rs-topology-requirements-00 (work in progress), February 2013.

[I-D.penno-sfc-yang]
Penno, R., Quinn, P., Zhou, D., and J. Li, "Yang Data Model for Service Function Chaining", draft-penno-sfc-yang-13 (work in progress), March 2015.

[I-D.wang-i2rs-rib-data-model]
Wang, L., Ananthakrishnan, H., Chen, M., amit.dass@ericsson.com, a., Kini, S., and N. Bahadur, "Data Model for RIB I2RS protocol", draft-wang-i2rs-rib-data-model-02 (work in progress), March 2015.

[I-D.xia-sfc-yang-oam]
Xia, L., Wu, Q., Kumar, D., Boucadair, M., and Z. Wang, "YANG Data Model for SFC Operations, Administration, and Maintenance (OAM)", draft-xia-sfc-yang-oam-02 (work in progress), March 2015.

[NFV-Terminology]
"Network Functions Virtualization (NFV); Terminology for Main Concepts in NFV", <http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.01.01_60/gs_NFV003v010101p.pdf>.

Authors’ Addresses

Linda Dunbar
Huawei
6340 Legacy Drive, Suite 175
Plano, TX 75024
USA

Phone: +1-469-277-5840
Email: ldunbar@huawei.com

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com

Jeff Tantsuara
Ericsson

Email: jeff.tantsura@ericsson.com