

draft-mackie-sfc-using-virtual-networking-02

S. Mackie, B. Rijsman, Juniper Networks

M. Napierala, AT&T

D. Daino, Telecom Italia

D.R. Lopez, Telefonica I+D

D. Bernier, Bell Canada

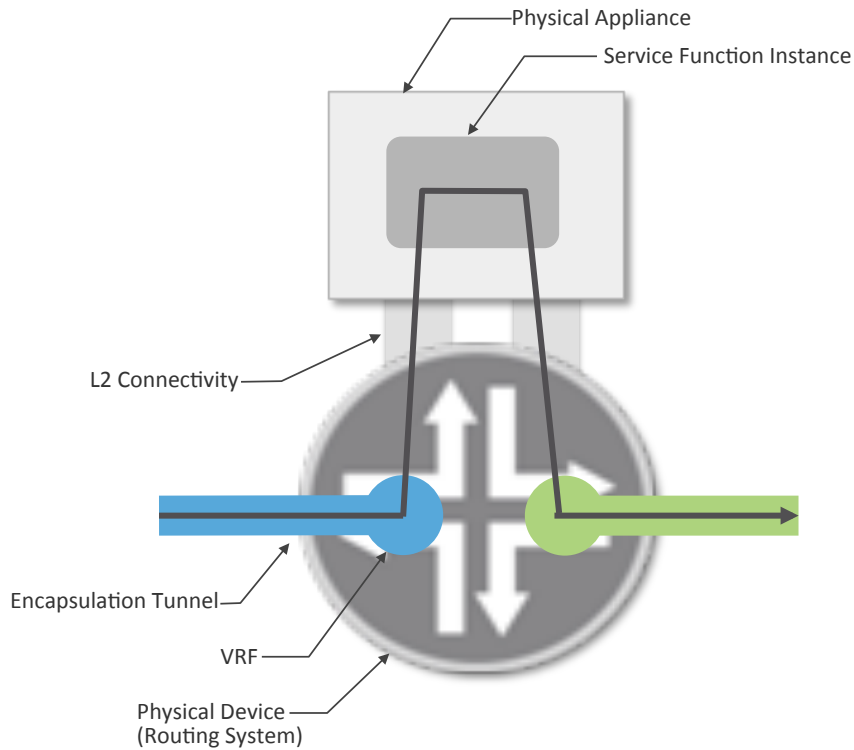
W. Haeffner, Vodafone

Goals of this Draft

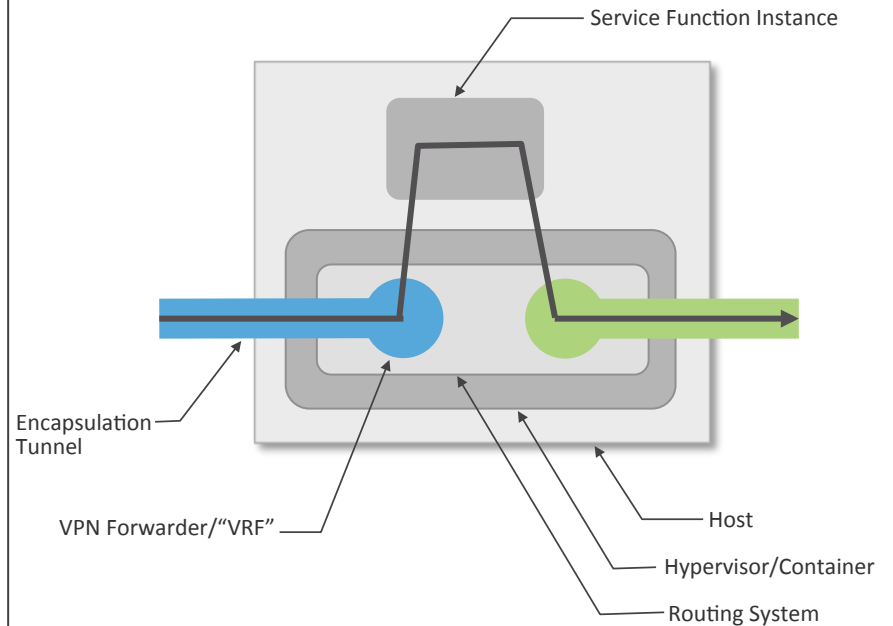
- Describe service chain implementation:
 - Controller to manage SFC topology and network configuration
 - Can use unmodified, existing protocols
 - Works with existing device capabilities
 - Supports physical/virtual forwarders and services
- Discuss load balancing
 - Flow stickiness
 - Forward/reverse symmetry

Service Instance Connection Detail

Physical Instance Connected to Physical Device



Virtualized Service Instance on a Host

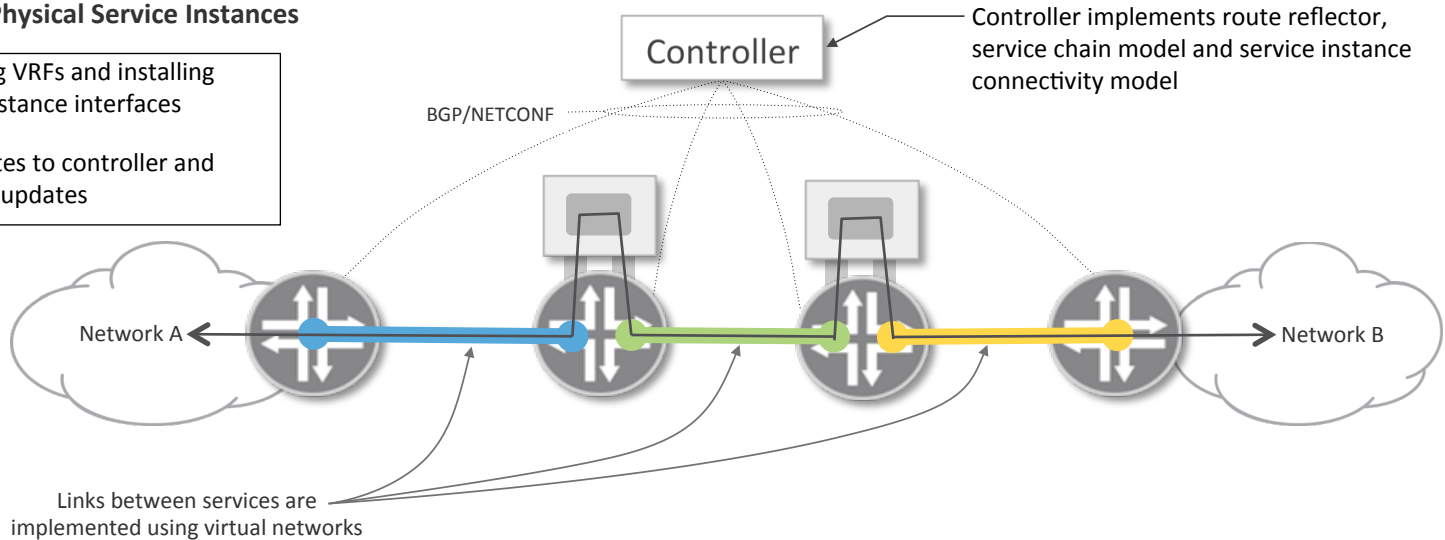


SFC-VN Architecture

Physical Devices with Physical Service Instances

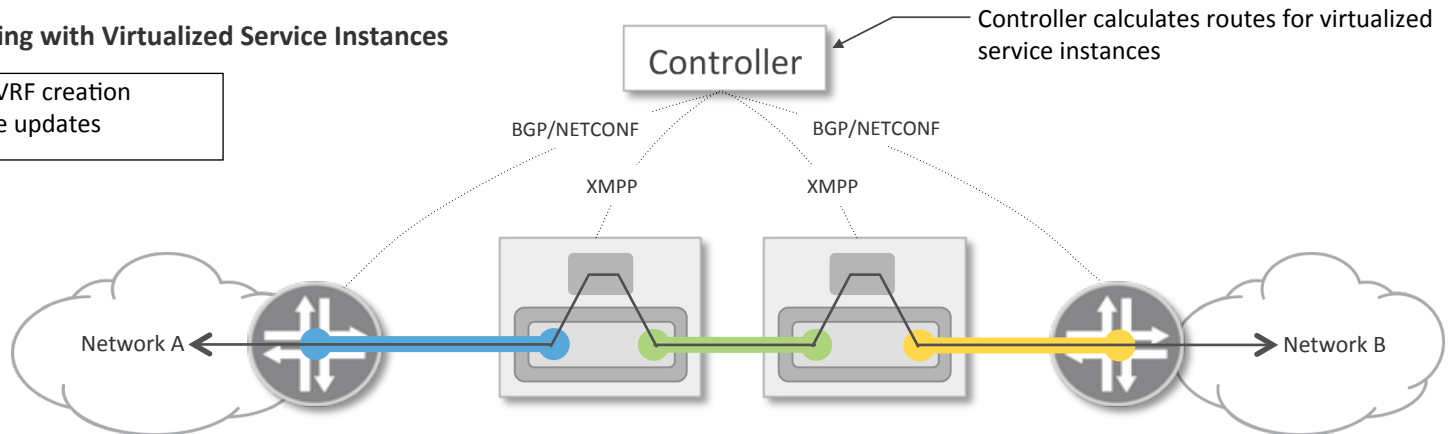
NETCONF for creating VRFs and installing routes into service instance interfaces

BGP to advertise routes to controller and controller to provide updates

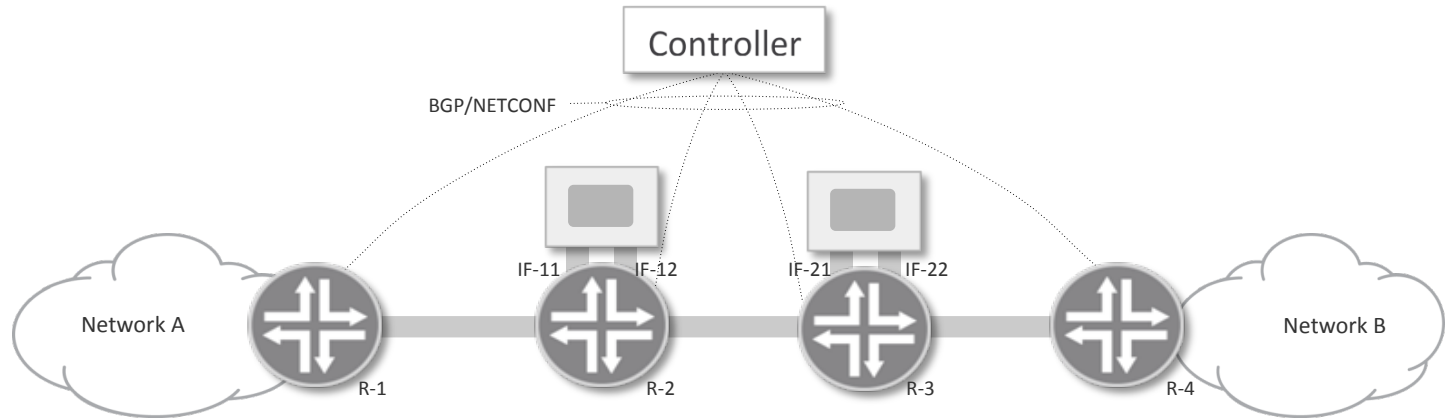


Server-Based Forwarding with Virtualized Service Instances

XMPP encapsulates VRF creation commands and route updates

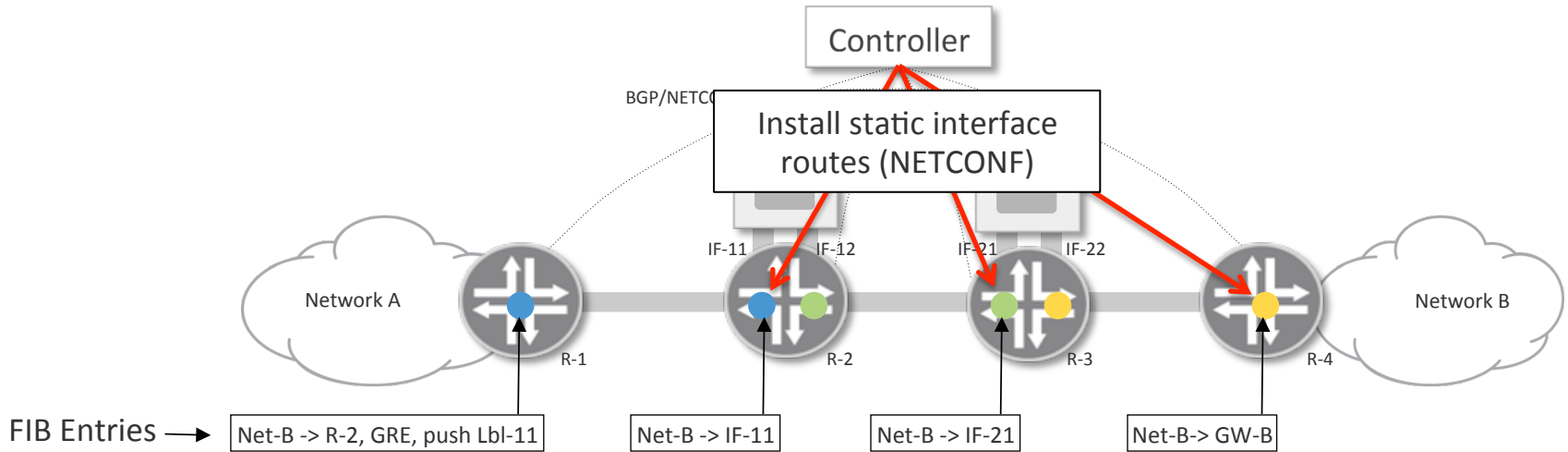


Creating a Service Chain



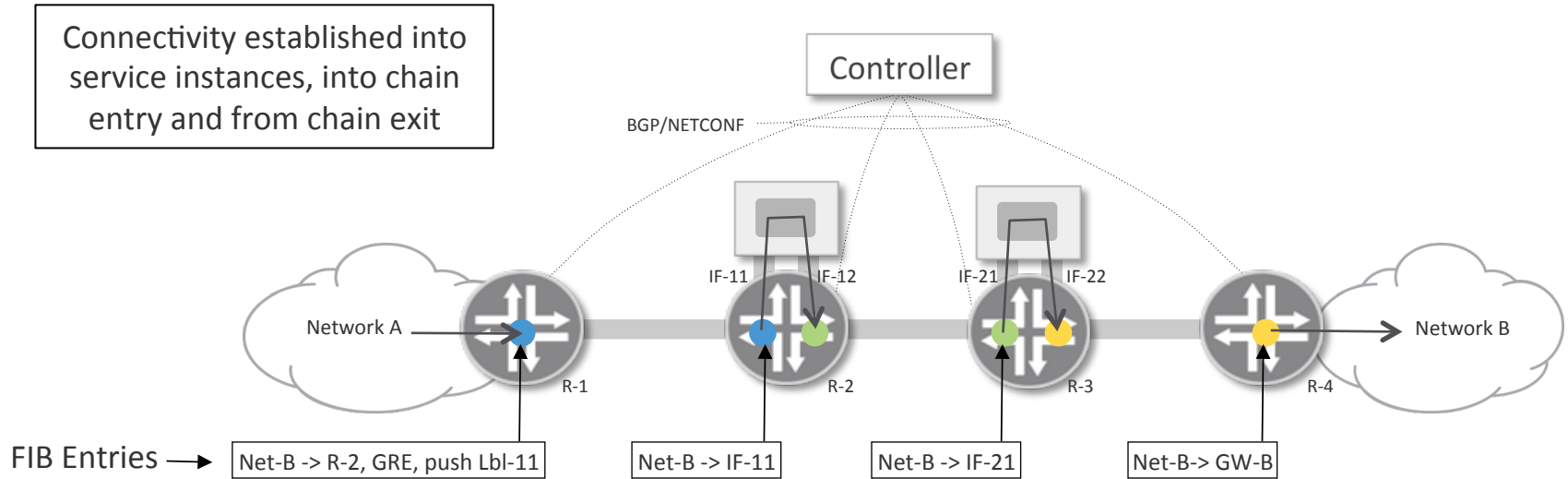
Initial physical/L2 connectivity

Creating a Service Chain



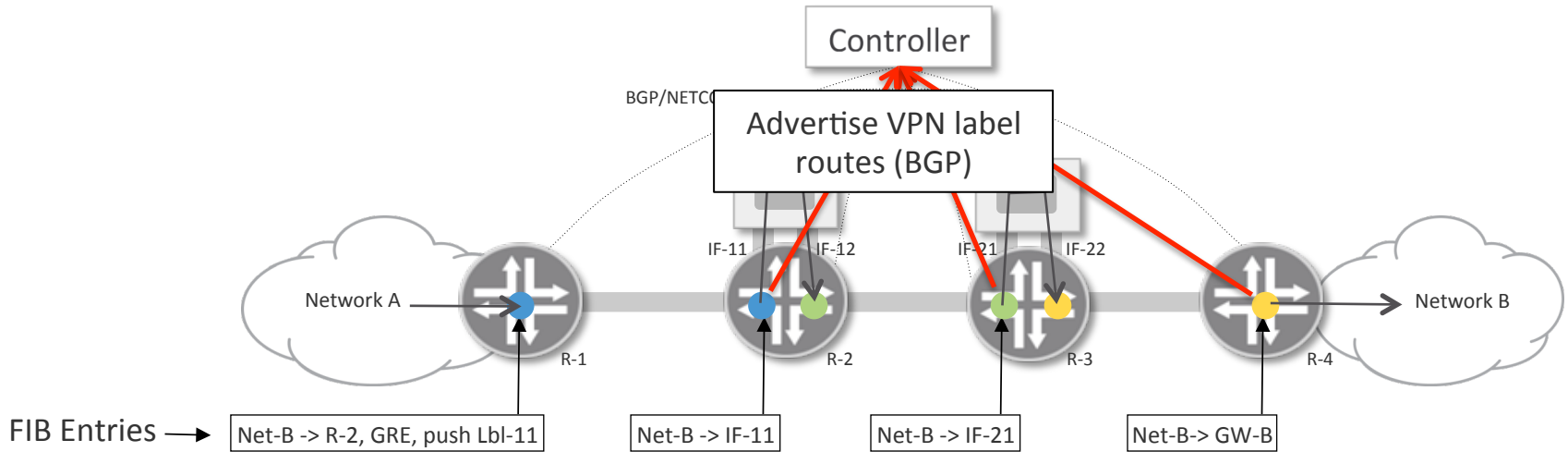
1. Configure VRFs containing each service instance interface, plus ingress/egress VRFs
2. Configure VRFs with import/export policies to define VPNs
3. Install static route in each ingress VRF for Network B pointing to service instance interface

Creating a Service Chain



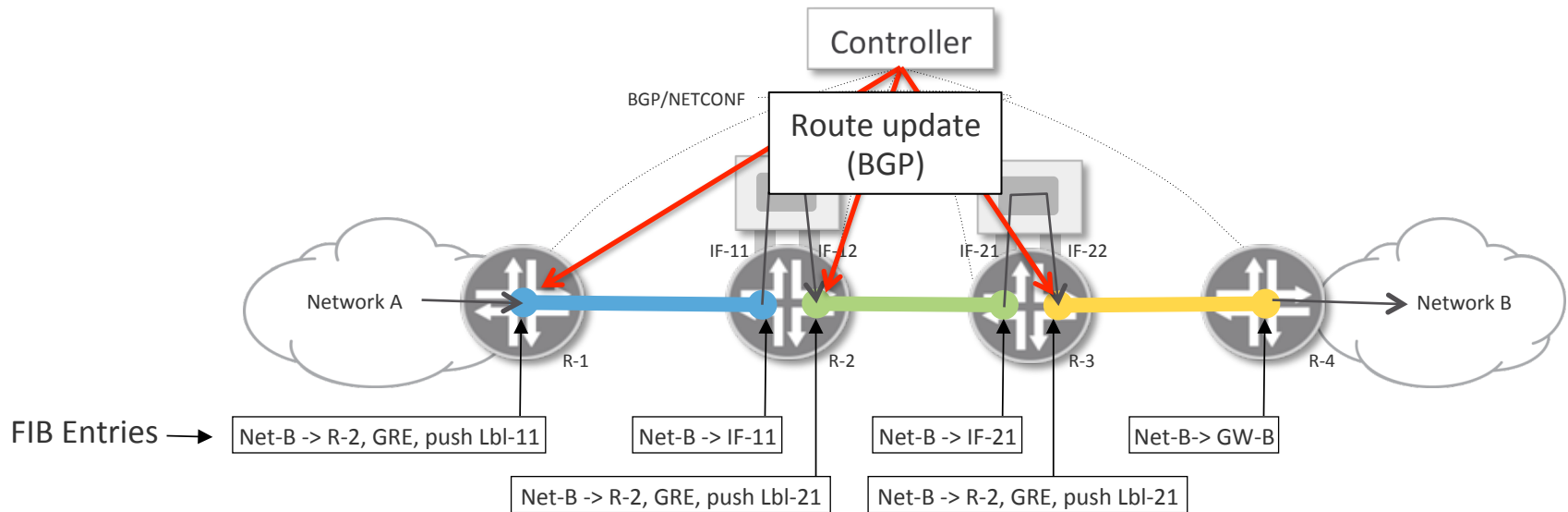
1. Configure VRFs containing each service instance interface, plus ingress/egress VRFs
2. Configure VRFs with import/export policies to define VPNs
3. Install static route in each ingress VRF for Network B pointing to service instance interface

Creating a Service Chain



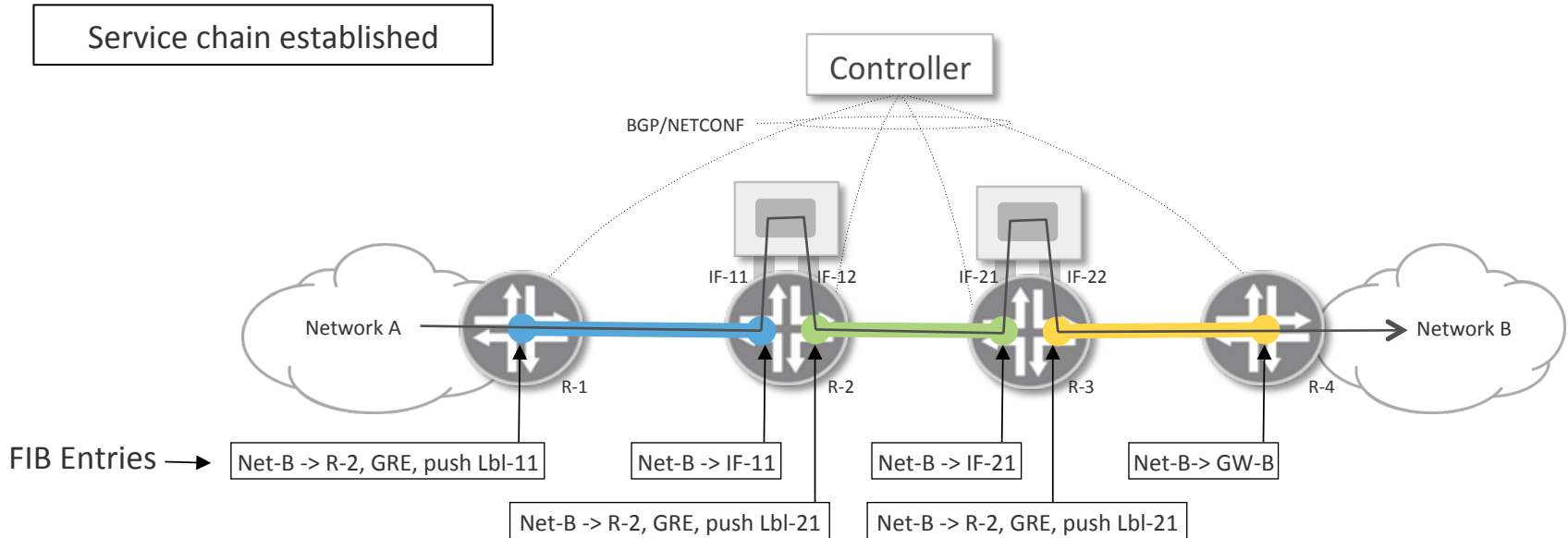
1. Configure VRFs containing each service instance interface, plus ingress/egress VRFs
2. Configure VRFs with import/export policies to define VPNs
3. Install static route in each ingress VRF for Network B pointing to service instance interface
4. Routers advertise labeled VPN routes

Creating a Service Chain



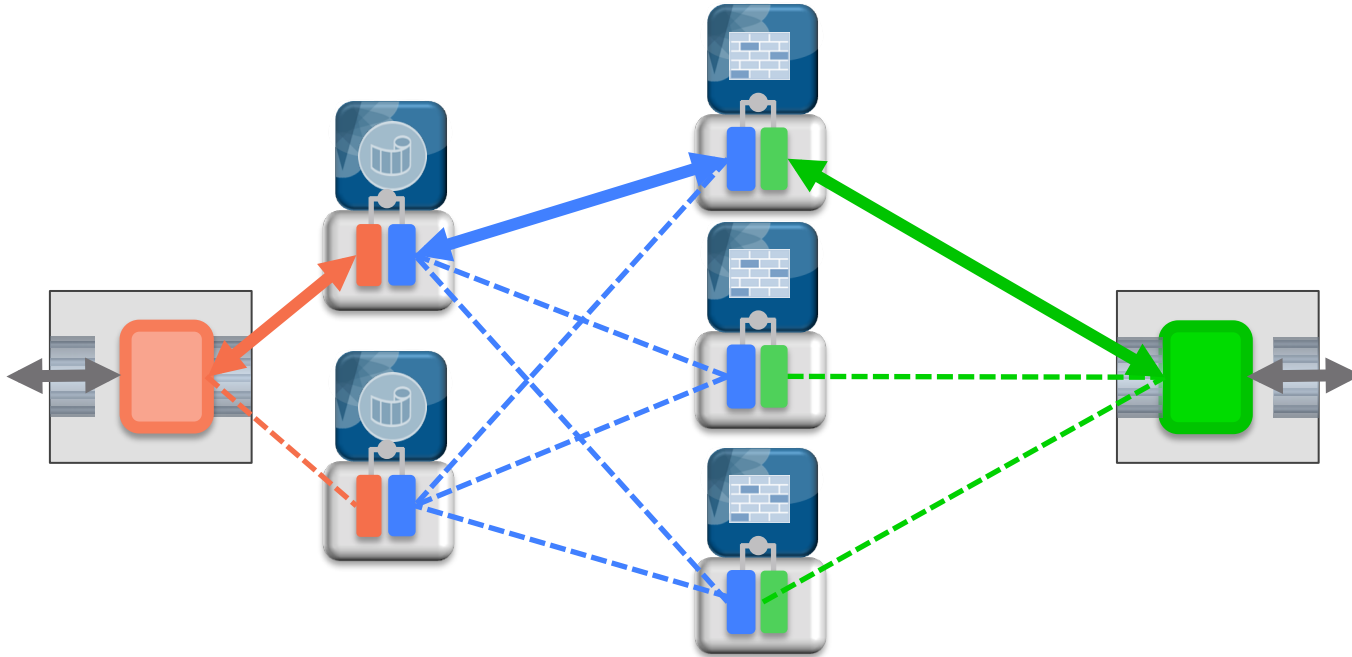
1. Configure VRFs containing each service instance interface, plus ingress/egress VRFs
2. Configure VRFs with import/export policies to define VPNs
3. Install static route in each ingress VRF for Network B pointing to service instance interface
4. Routers advertise labeled VPN routes
5. Controller provides updates to VRFs with matching communities to create connection between egress of one service instance to ingress of next.

Creating a Service Chain



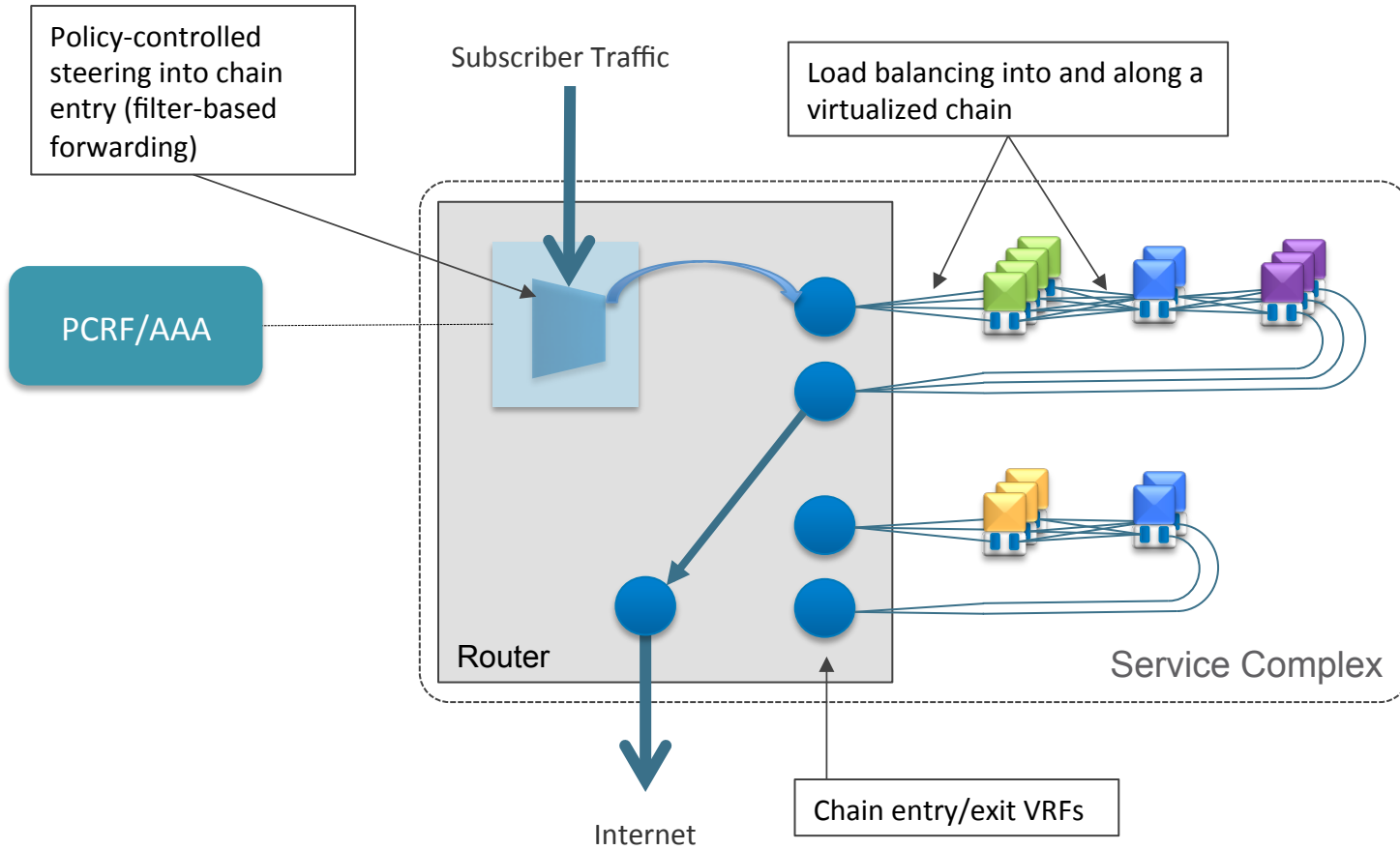
1. Configure VRFs containing each service instance interface, plus ingress/egress VRFs
2. Configure VRFs with import/export policies to define VPNs
3. Install static route in each ingress VRF for Network A pointing to service instance interface
4. Routers advertise labeled VPN routes
5. Controller provides updates to VRFs with matching communities to create connection between egress of one service instance to ingress of next.

Load Balancing



- Load balance in VRFs attached to service instances
- Learned flow tables and/or consistent hash to avoid rehash during service scaling
- Ensure reverse hash same as forward hash for flow symmetry through service instances
- Scale out on same server needs label-based hashing, or BGP link bandwidth extended community

Classification



Other Items

- Reclassification
 - Support multiple egress interfaces in an SF
 - Classify by 5-tuple in VRFs
- Metadata
 - Will work transparently with service metadata
 - Can work with network metadata (GENEVE)
 - Service to network interaction handled via controller (similar to 3GPP/PCRF)