Building Virtual Service Topologies in BGP VPNs
(draft-rfernando-virt-topo-bgp-vpn)

Dhananjaya Rao
Rex Fernando
Luyuan Fang
Cisco

IETF 85, November 2012
Atlanta
Introduction

- Traditional networks achieve isolation and services within a VPN using topological constraints and routing
  - Services applied at service nodes placed in traffic path
- L3VPN supports constrained policy based routing
  - Extranecls, hub-n-spoke mechanisms used to re-direct traffic for services
- Porting traditional network service model to a virtualized data center requires:
  - Create more complex topologies for each tenant VPN
  - Flexibly and efficiently constrain the flow of routes and traffic over these virtual topologies
- L3VPN constructs can be easily extended to support flexible service chains
Purpose of the draft

- Describe the concept of Virtual Service Topologies in BGP VPNs
  - Defined using standard VPN constructs
- Describe functions at a PE to efficiently constrain routing and traffic flows over these topologies
- Informational
Virtual Machine Intra-zone routing

- In a data center, servers host virtual machines where end applications reside
  - Each application VM is a CE from a BGP VPN perspective
- A collection of CE/VMs that can communicate freely form a zone
  - A PE creates a VRF for its attached CE/VMs in a zone
- Intra-zone connectivity achieved by designating a RT per zone (zone-RT)
  - Applied on all PE VRFs that terminate the CE/VMs that belong to the zone
Inter-zone Routing and Traffic Forwarding

- Inter-zone traffic may need the ability to apply network policies and services in a specific order
- Service nodes may be VMs spread across the data center
- Inter-zone traffic must follow a predetermined service path and forwarding through one or more service nodes
- A sequence of service-PEs and their attached service nodes creates a unidirectional service chain or topology
- Two step process:
  - Virtual Service Topology construction
  - Inter-zone Routing and Service Chaining
Inter-zone Routing
Service Topology Route and Zone prefixes

As Service-PE₁ has no import-RT for the service-topology-route associated with Service-topology-RT:100:1 it becomes the Destination Service-PE and generates the service-topology-route with itself as next-hop.

Service-PE₁

VRF

FIB/LIB

192.168.1.1/32, NH PE₂, VPN label (123)

Generate: Service-topology-route (100.1.1.1/32) VPN Out_label (123), VPN In_label (192) -> Service Node₂

Export: Service-export-RT: 100:2
Import: Service-topology-RT:100:1

Service VRF

FIB/LIB

192.168.1.1/32, NH 100.1.1.1/32, VPN label (123)

Reoriginate: Service-topology-route (100.1.1.1/32) NH=PE₁, VPN Out_label (192), VPN In_label (127) -> Service Node₂

Import: Service-import-RT: 100:2
Export: Service-export-RT: 100:3
Import: Service-topology-RT:100:1

PE₁ has no export-RT for the service-topology-route associated with Service-topology-RT:100:1 and becomes the Source Service-PE. It rewrites the next-hop in FIB for any zone-prefix that carries the service-topology-RT to the service-topology-route.

Service-PE₂

VRF

FIB/LIB

192.168.1.1/32, NH 100.1.1.1/32, VPN label (123)

Service-topology-route (100.1.1.1/32) NH=PE₁, VPN Out_label (127)

Import: Service-import-RT: 100:3
Import: Service-topology-RT:100:1

Service PE₂ is not the Destination Service-PE as it has an import statement associated with the service-topology-route. It therefore rewrites the next-hop in FIB for any zone-prefix that carries the service-topology-RT to the associated service-topology-route, and re-originate the service-topology-route with itself as next-hop.

Service Node₁

Service Node₂

Zone Prefix = 192.168.1.1/32, Next-hop = PE₀, VPN Label (123)
Service-topology-RT: 100:1 Zone-RT: green-zone

As Service-PE₃, has no import-RT for the service-topology-route associated with Service-topology-RT:100:1 it becomes the Destination Service-PE and generates the service-topology-route with itself as next-hop.

Service-PE₃

VRF

FIB/LIB

192.168.1.1/32

Generate: Service-topology-route (100.1.1.1/32) VPN Out_label (123), VPN In_label (192) -> Service Node₂

Import: Service-import-RT: 100:2
Import: Service-topology-RT:100:1

Service VRF

FIB/LIB

192.168.1.1/32, NH 100.1.1.1/32, VPN label (123)

Reoriginate: Service-topology-route (100.1.1.1/32) NH=PE₁, VPN Out_label (192), VPN In_label (127) -> Service Node₂

Import: Service-import-RT: 100:2
Export: Service-export-RT: 100:3
Import: Service-topology-RT:100:1

Service-PE₁ has no export-RT for the service-topology-route associated with Service-topology-RT:100:1 and becomes the Source Service-PE. It rewrites the next-hop in FIB for any zone-prefix that carries the service-topology-RT to the service-topology-route.

Service-PE₁

VRF

FIB/LIB

192.168.1.1/32, NH 100.1.1.1/32, VPN label (123)

Service-topology-route (100.1.1.1/32) NH=PE₁, VPN Out_label (127)

Import: Service-import-RT: 100:3
Import: Service-topology-RT:100:1

Service VRF

FIB/LIB

192.168.1.1/32, NH 100.1.1.1/32, VPN label (123)

Service-topology-route (100.1.1.1/32) NH=PE₂, VPN Out_label (127)

Import: Service-import-RT: 100:3
Import: Service-topology-RT:100:1
Inter-zone Routing
Inter-zone Traffic Forwarding

Zone: 1

Service Node 1

FIB/LIB

PE 1

Service VRF

VRF 1

Zone: 2

Service Node 2

FIB/LIB

PE 2

Service VRF

VRF 2

Zone: 3

Service Node 3

FIB/LIB

PE 3

Service VRF

VRF 3

192.168.1.1/32, VPN In Label (123)
192.168.1.1/32, NH PE 0, VPN label (123)
100.1.1.1/32, VPN Out_label (123), VPN In_label (123)
100.1.1.1/32) NH=SPE 1, VPN Out_label (192), VPN In_label (127)
192.168.1.1/32, NH 100.1.1.1/32, VPN label (123)
100.1.1.1/32) NH=SPE 1, VPN Out_label (192), VPN In_label (127)

Dst: 192.168.1.1/32

192.168.1.1

192.168.1.1

192.168.1.1
Next Steps

-01 version will have detailed descriptions with examples
Describe additional scenarios and optimizations
Incorporate comments and feedback