Configure Basic Features

SD-Fabric Tutorial – Part 2

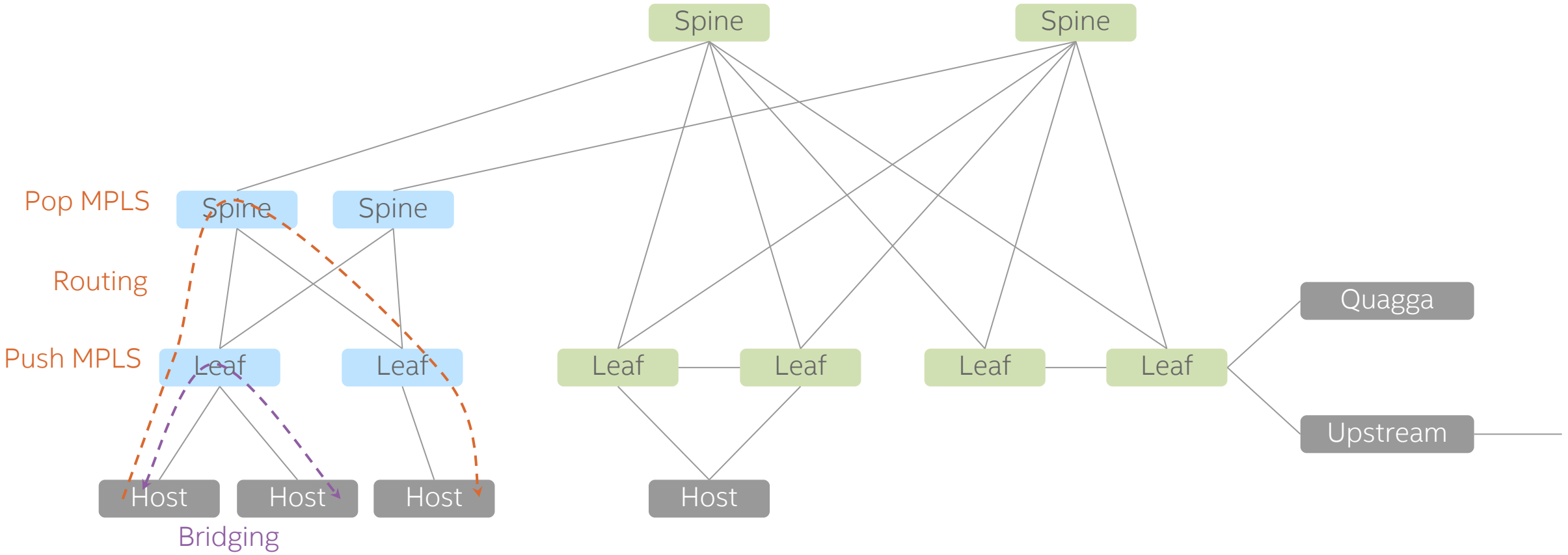
Forwarding Features

Commonly seen in commercial solutions

Forwarding Features

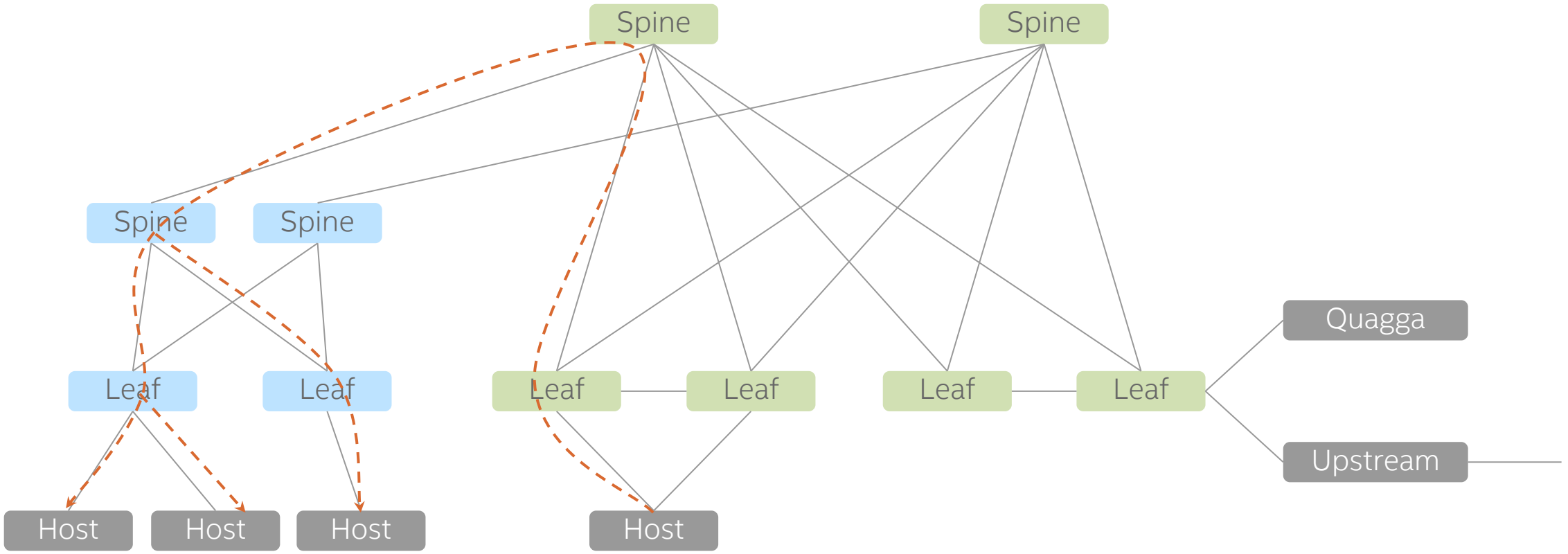
- Bridging with Access & Trunk VLANs (within a rack)
- Routing (inter-rack)
  - IPv4 & IPv6 Unicast routing with MPLS Segment-Routing
  - IPv4 & IPv6 Multicast routing
- Dual-homing for compute-nodes and external routers
- Multi-stage fabrics (2 layers of spines)
- Virtual Router (vRouter) - entire fabric behaves as a single router
  - BGP (v4/v6) support for external connectivity
  - Static routes, route blackholing
- DHCP L3 relay (v4/v6)

Bridging & Routing

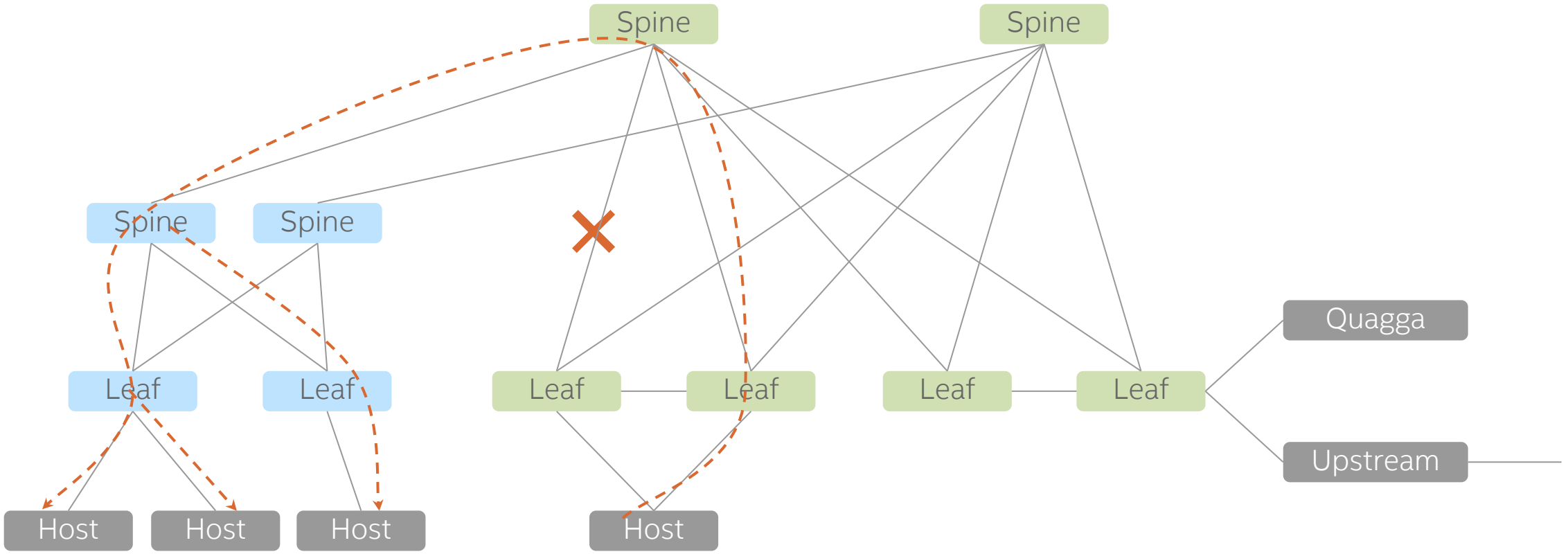


Quagga: a routing software suite, providing implementations of various routing protocols

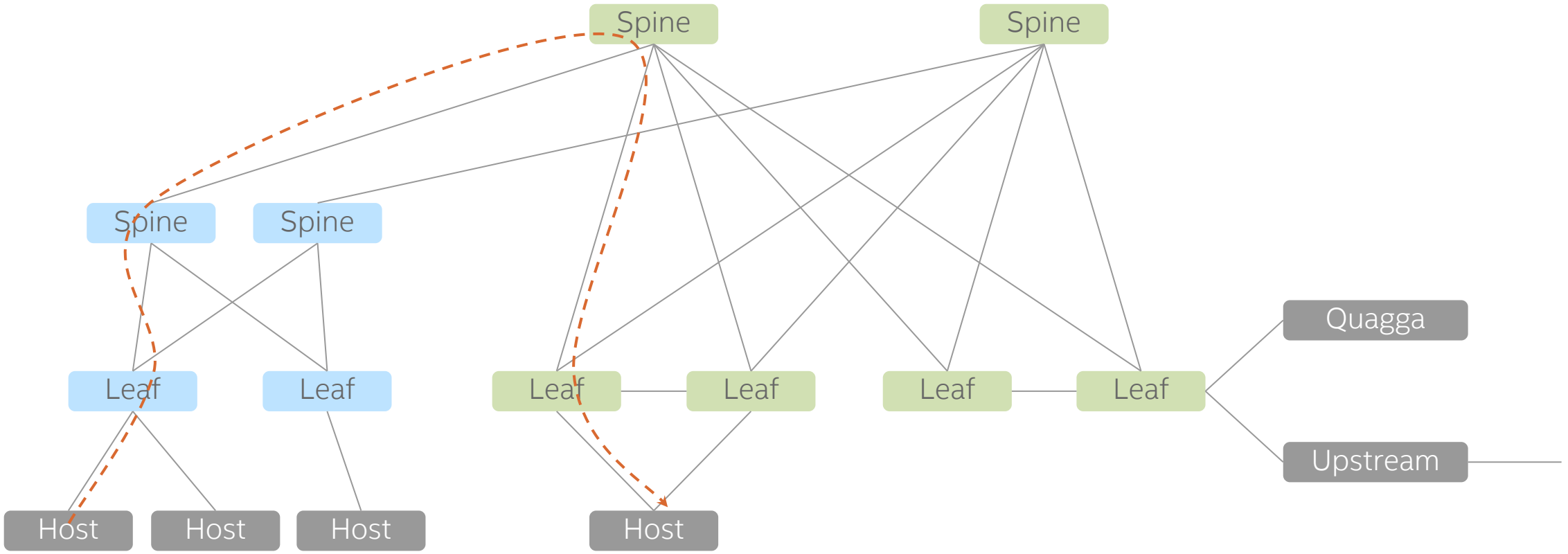
Multicast


Spine
Spine
Spine
Spine
Leaf
Leaf
Leaf
Leaf
Leaf
Quagga
Upstream
Host
Host
Host
Host

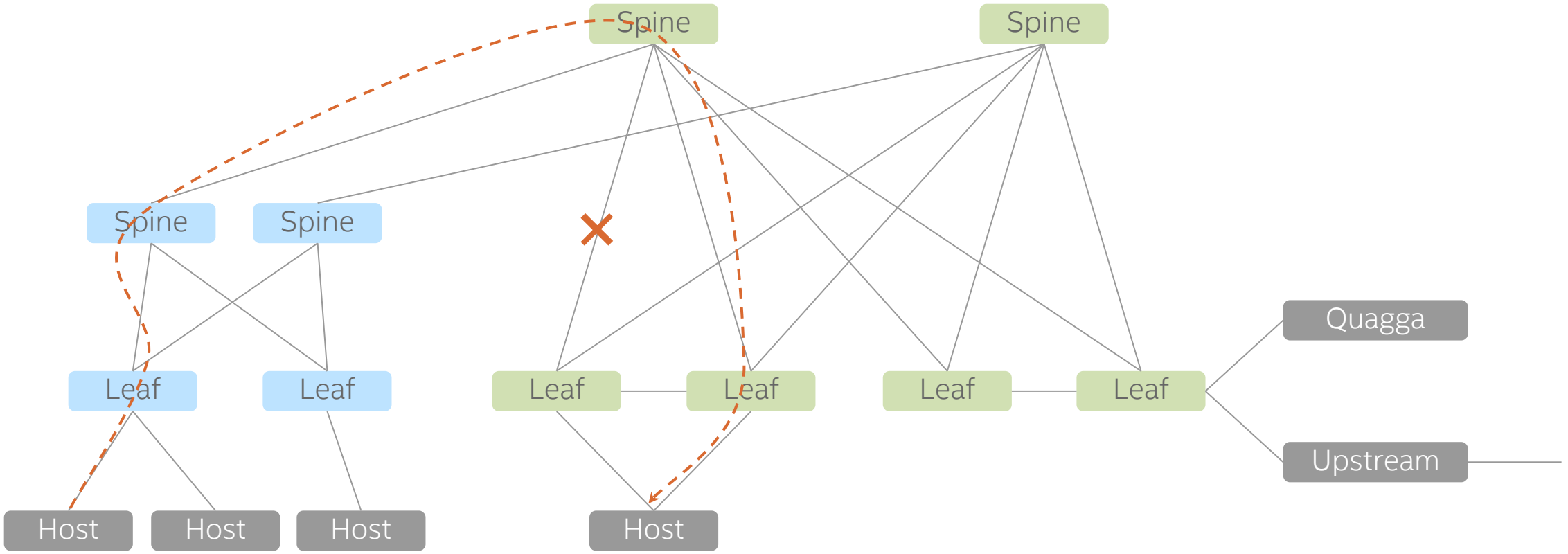
Multicast (failure)



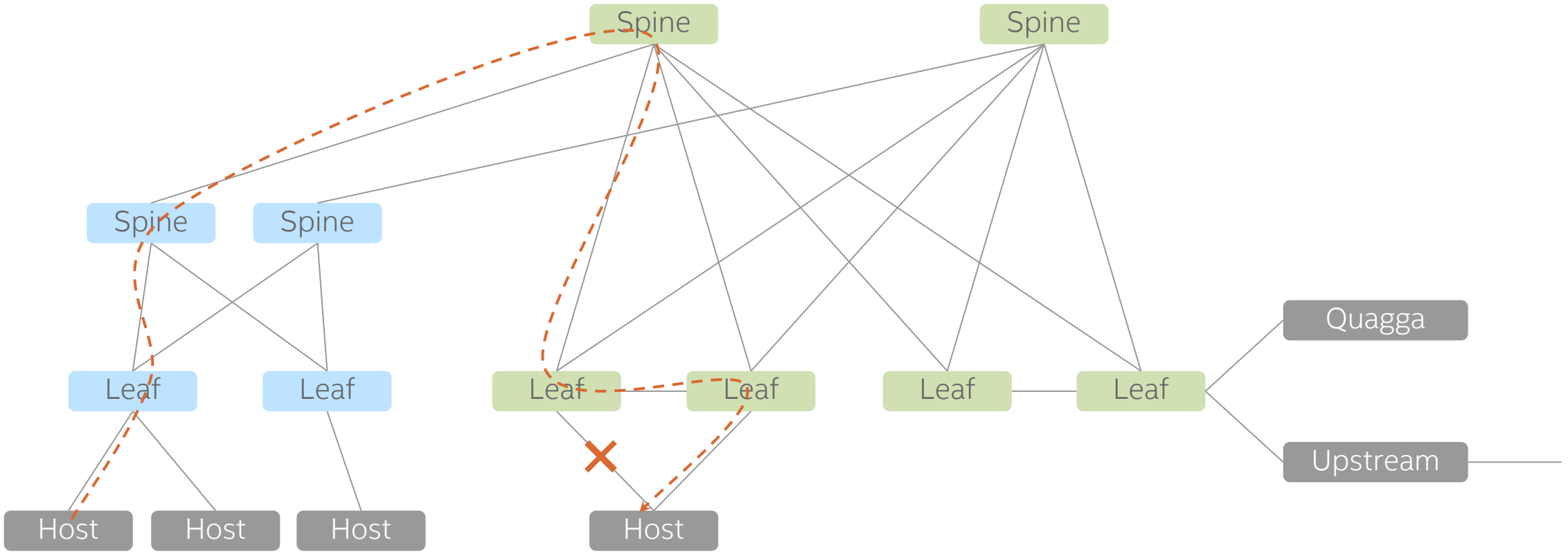
Dual-Homing


Spine
Spine
Spine
Spine
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Quagga
Upstream
Host
Host
Host
Host

Dual-Homing (global failure)

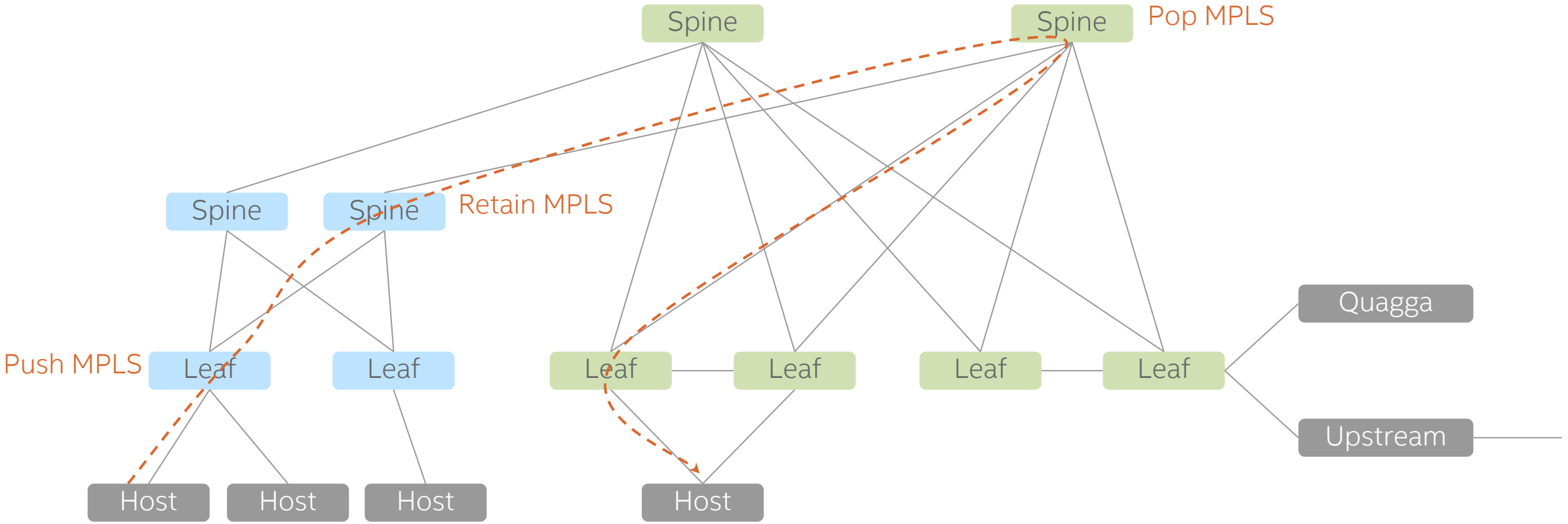


Dual-Homing (local failure)

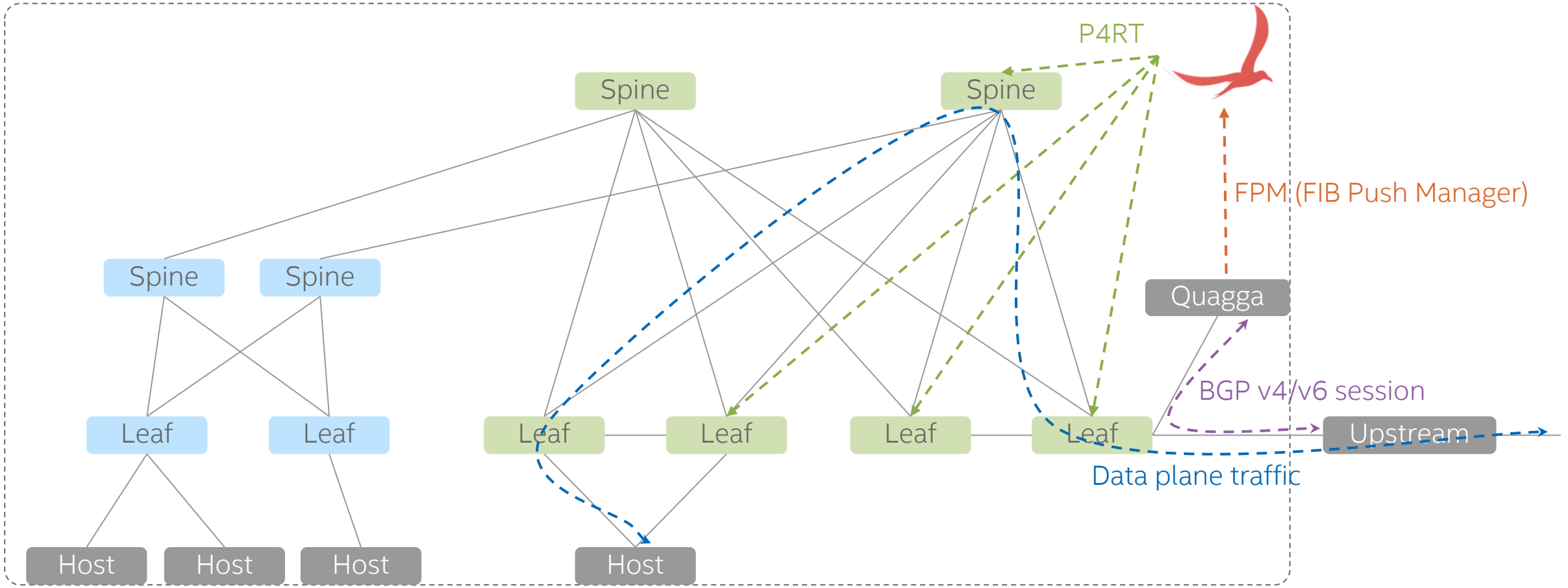


Pair link is only used to recover local failure

Multi-Stage

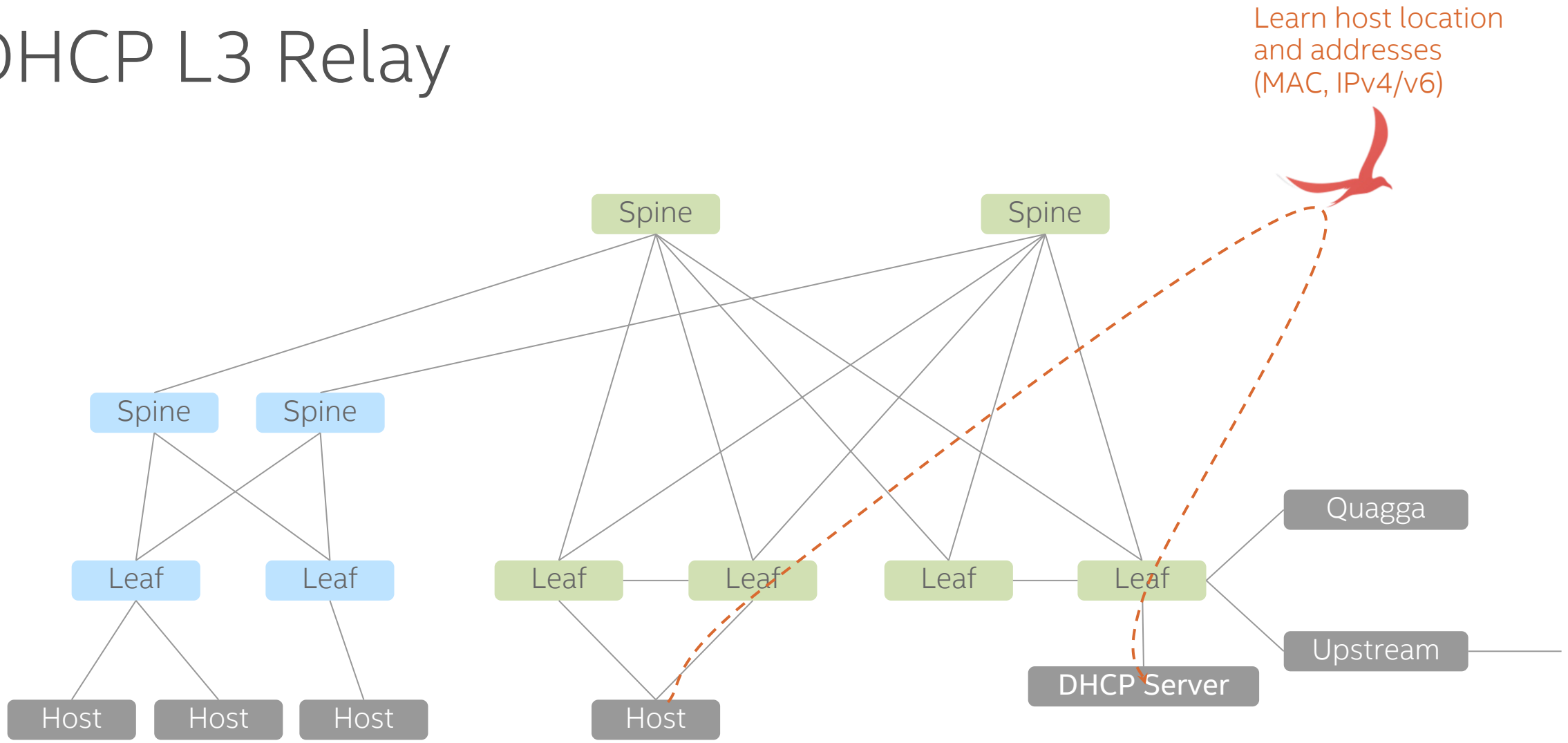

Spine
Spine
Pop MPLS
Spine
Spine
Retain MPLS
Push MPLS
Leaf
Leaf
Leaf
Leaf
Leaf
Leaf
Quagga
Upstream
Host
Host
Host
Host

vRouter



- FPM: deliver forwarding plane information in Quagga to ONOS
- The entire SD-Fabric is abstracted as one big router to the outside
- Control / data separation

DHCP L3 Relay



Control Plane Software Components

SDN applications that implements the features

SD-Fabric Applications

Mandatory

- drivers
  drivers for various devices and pipelines
- segmentrouting
  controls forwarding in the fabric
- hostprovider, lldpprovider

Mandatory for P4 switches

review session 3 for details

- drivers.bmv2 or drivers.stratum-tofino
  - drivers.stratum
    - generaldeviceprovider
    - drivers.p4runtime
    - drivers.gnmi
    - drivers.gnoi
    - pipelines.basic
- org.stratumproject.fabric-tna
  P4 program for both v1Model (BMV2) and TNA (Tofino)
  - protocols.p4runtime
    - protocols.grpc

Optional

- gui2
  enables graphic user interface. Highly recommended
- fpm (Forwarding Plane Manager)
  exchanges forwarding information with Quagga
- route-service
  route store and API
- mcast
  multicast store and API
- dhcprelay
  relays DHCP packets between clients and servers
- routeradvertisement
  periodically sends IPv6 router advertisement packets on configured interfaces
- hostprobingprovider
  probes and verifies locations of dual-homed hosts
- netcfghostprovider
  allow static host configuration
- org.omecprojecet.up4
  5G UPF control app

Configuration

ONOS network configuration (netcfg)

Overview

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

```
  "ports": {
    "device:leaf1/3": {
      "interfaces": [
        {
          "name": "leaf1-3",
          "ips": [
            "172.16.1.254/24"
          ],
          "vlan-untagged": 100
        }
      ]
    },
    "device:leaf1/4": {
      ...
    },
    "device:leaf1/5": {
      ...
    },
    "device:leaf1/6": {
      ...
    }
  }
}
```

- One “devices” config per device.
- One “ports” config per edge port and pair port
  - No need to configure infra port

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

ID of the switch

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

Management address of Stratum agent

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

Device driver and pipeconf

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

For UI display only

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

User friendly name that will be displayed on UI

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

Segment ID.
Can be an arbitrary value but need to be unique in the system.
Also used as the MPLS label when doing segment routing.
Do not use reserved MPLS labels (i.e. <=16).

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

Loopback address of the switch.
Can be an arbitrary value (e.g. the management IP) but need to be unique in the system.

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

Used for ARP reply when host ARPs its gateway.
Can be an arbitrary value (e.g. the management MAC) but need to be unique in the system.

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

True for leaves. False for spines.

Device Configuration

```
{
  "devices": {
    "device:leaf1": {
      "basic": {
        "managementAddress": "grpc://mininet:50001?device_id=1",
        "driver": "stratum-bmv2",
        "pipeconf": "org.stratumproject.fabric-upf.bmv2",
        "locType": "grid",
        "gridX": 200,
        "gridY": 600,
        "name": "leaf1"
      },
      "segmentrouting": {
        "ipv4NodeSid": 101,
        "ipv4Loopback": "192.168.1.1",
        "routerMac": "00:AA:00:00:00:01",
        "isEdgeRouter": true,
        "adjacencySids": []
      }
    },
    "device:leaf2": {
      ...
    },
    "device:spine1": {
      ...
    },
    "device:spine2": {
      ...
    }
  }
```

Deprecated. Just put an empty array for now.

Port Configuration - VLAN untagged and tagged

```
"ports": {
"device:leaf1/3": {
  "interfaces": [
    {
      "name": "leaf1-3",
      "ips": [
        "172.16.1.254/24"
      ],
      "vlan-untagged": 100
    }
  ]
},
"device:leaf1/4": {
  ...
},
"device:leaf1/5": {
  "interfaces": [
    {
      "name": "leaf1-5",
      "ips": [
        "172.16.1.254/24"
      ],
      "vlan-tagged": [
        100
      ]
    }
  ]
},
...
```

Device ID/port number

User friendly name

Equivalent to **access port**.
Expecting untagged packet and will push VLAN 100 inside the switch

Equivalent to **trunk port**.
Expecting VLAN 100 tagged packets and will keep the tag.
It is possible to have more than one trunk VLAN.

Port Configuration - VLAN native

```
  "ports": {
  "device:leaf1/9": {
    "interfaces": [
      {
        "name": "leaf1-9",
        "ips": [
          "172.16.9.254/24"
        ],
        "vlan-native": 800,
        "vlan-tagged": [
          900
        ]
      }
    ]
  },
  ...
 }
}
```

Note: exercise 1 doesn't use native VLAN.
A non-existing port is used in order to explain how this works.

Equivalent to trunk port with native VLAN.
Expecting VLAN 900 tagged packet and will keep the tag.
Also expecting untagged packet and will push VLAN 800

Port Configuration - Subnet

```
"ports": {
"device:leaf1/3": {
  "interfaces": [
    {
      "name": "leaf1-3",
      "ips": [
        "172.16.1.254/24"
      ],
      "vlan-untagged": 100
    }
  ]
},
"device:leaf1/4": {
  ...
},
"device:leaf1/5": {
  "interfaces": [
    {
      "name": "leaf1-5",
      "ips": [
        "172.16.1.254/24"
      ],
      "vlan-tagged": [
        100
      ]
    }
  ]
},
...
```

Two information can be derived from the subnet config:
(1) The interface IP on the switch is 172.16.1.254
(2) The subnet on this interface is 172.16.1.0/24

Environment Overview

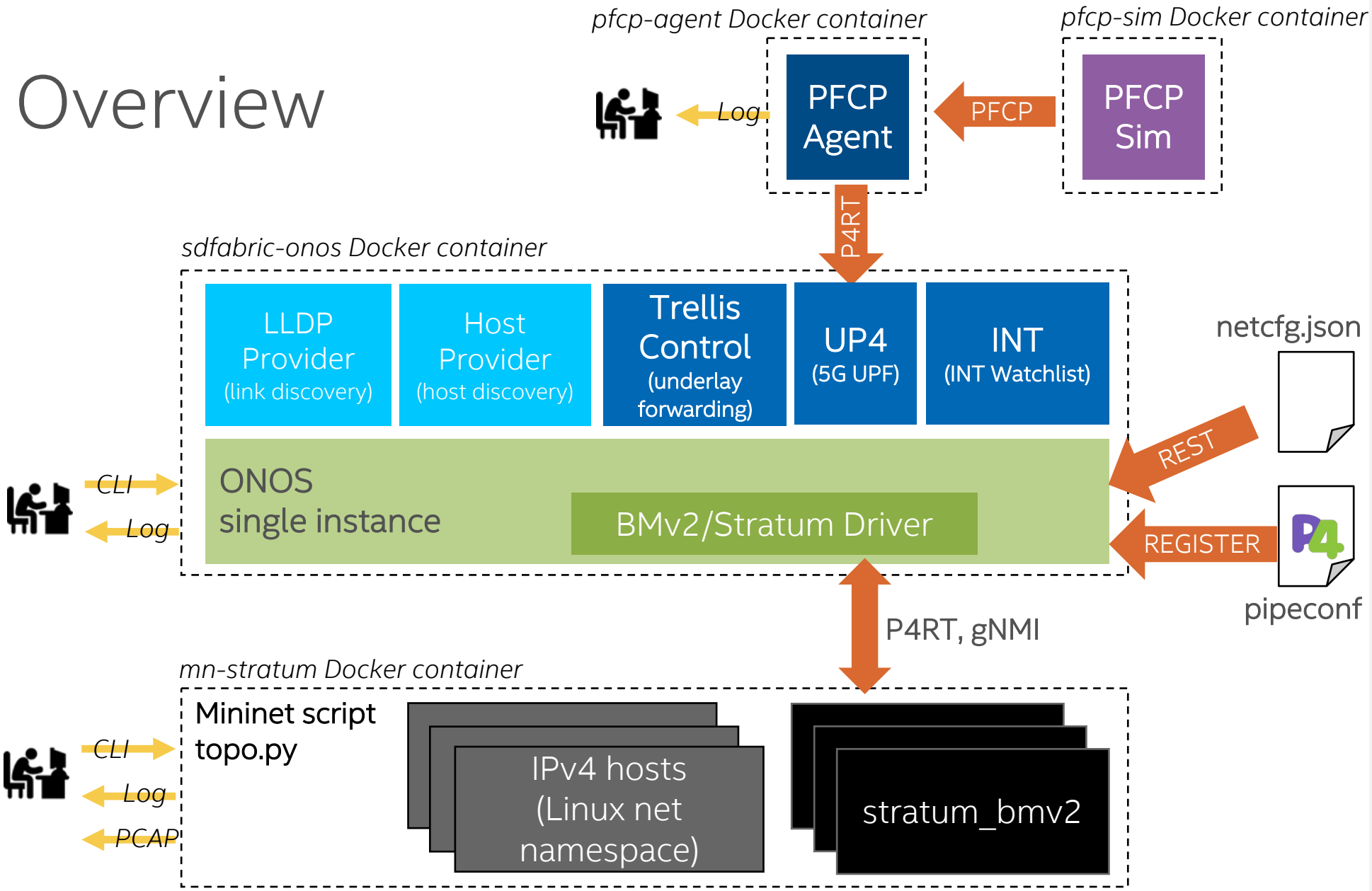
Recommended Background Knowledge

- Visit Part 0~4 of NG-SDN Tutorial to learn the basics of
  - ONOS
  - Stratum
  - Control protocols (P4Runtime, YANG, OpenConfig, gNMI)

Exercise Overview

Useful Commands

```
make deps
make start
make upf-start
make netcfg
make onos-cli
make onos-log
make mn-cli
make mn-log
make mn-pcap
make pfcp-log
```



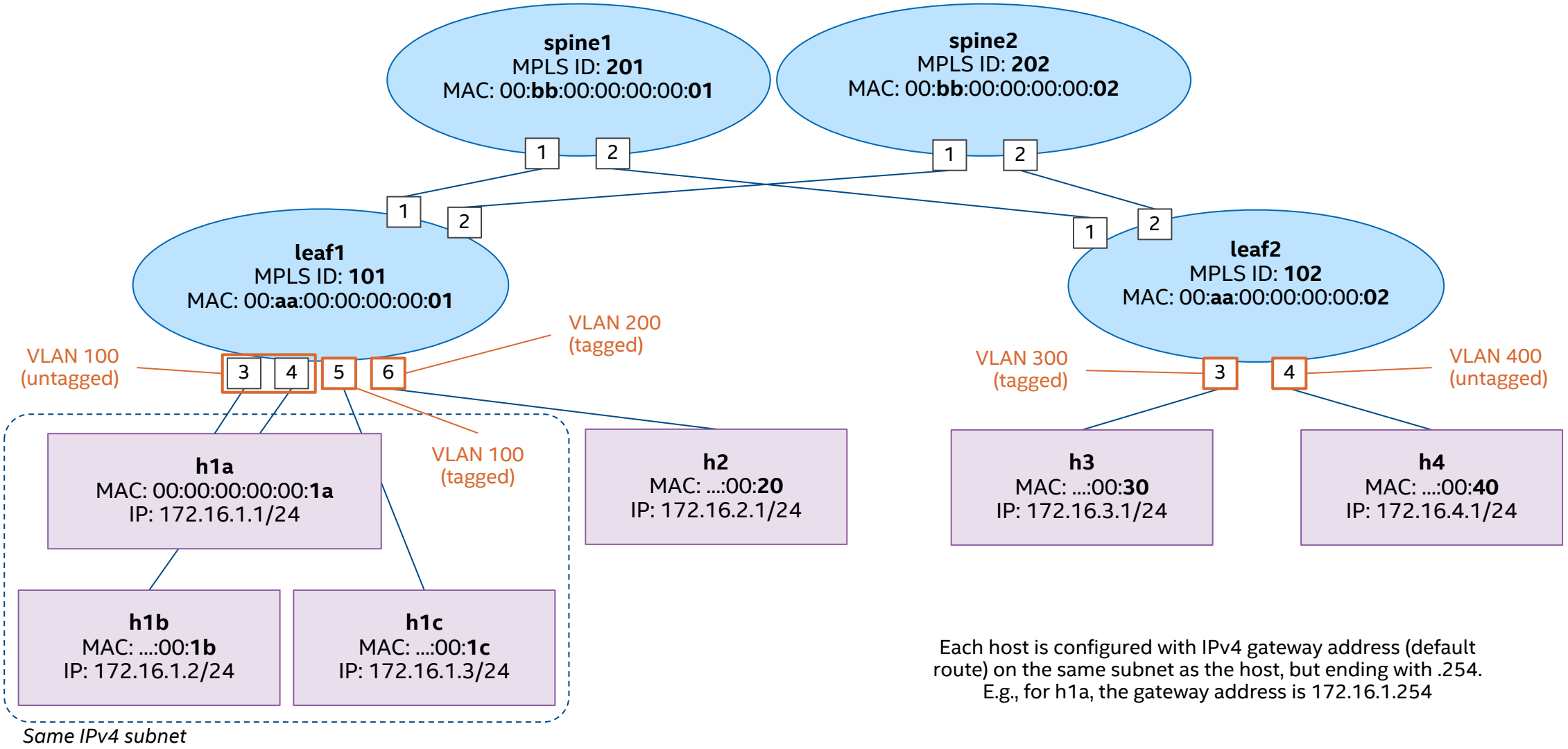
Exercise 1

SD-Fabric Basics

Exercise 1

- Basic operation
  - Start ONOS and Mininet with 2x2 topology and IPv4 hosts (make start)
  - Push given working config (make netcfg)
  - Verify connectivity
- Modify config
  - Some hosts cannot be pinged!
  - Add extra interface config
  - Verify connectivity of the extra host

Mininet Topology for Exercise 1



Each host is configured with IPv4 gateway address (default route) on the same subnet as the host, but ending with .254. E.g., for h1a, the gateway address is 172.16.1.254

Exercise 1: Get Started

- Open lab README on GitHub
  - http://github.com/opennetworkinglab/sdfabric-tutorial
- Or open in text editor
  - sdfabric-tutorial/README.md
  - sdfabric-tutorial/EXERCISE-1.md
- Solution
  - sdfabric-tutorial/solution

Notices & Disclaimers

- Intel technologies may require enabled hardware, software or service activation.
- No product or component can be absolutely secure.
- Your costs and results may vary.
-


intel