

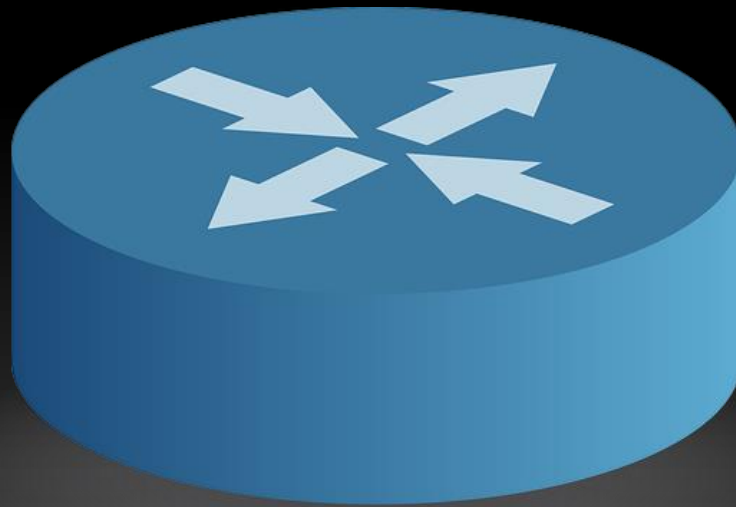
ODENOS : An SDN Framework for Heterogeneous Network Orchestration

Kazuya Suzuki

Cloud System Research Laboratories,
NEC Corporation

Jul/15/2015

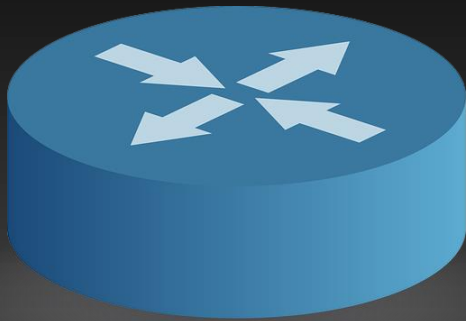
At the time of a firmware update,
I want to move data flows on the device to
another path.



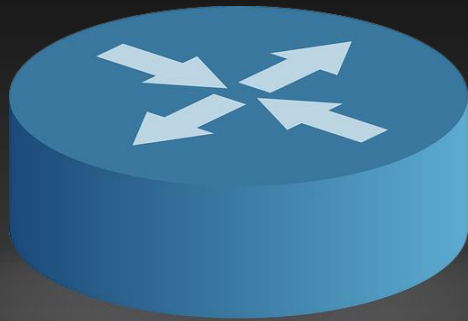
As communication quality gets wrong,
I want to use another path.



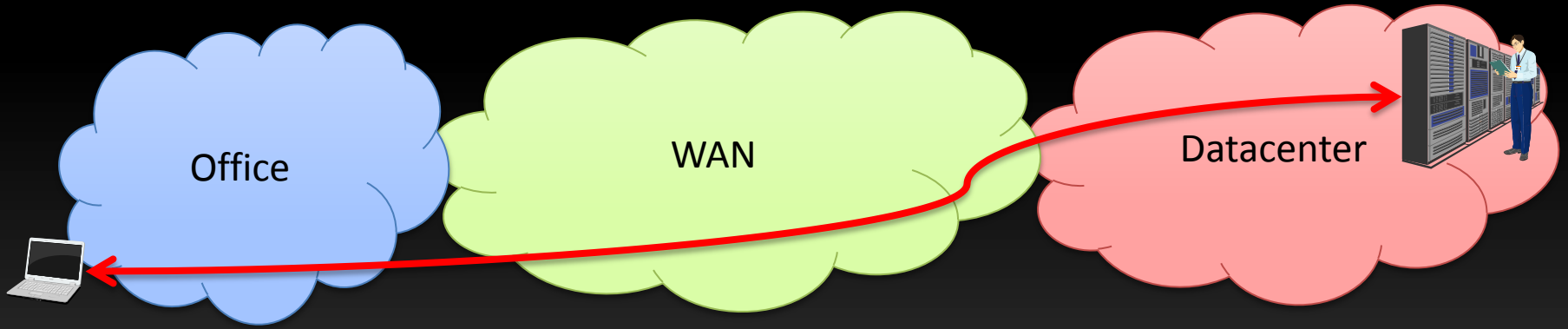
Network Visualization and Control

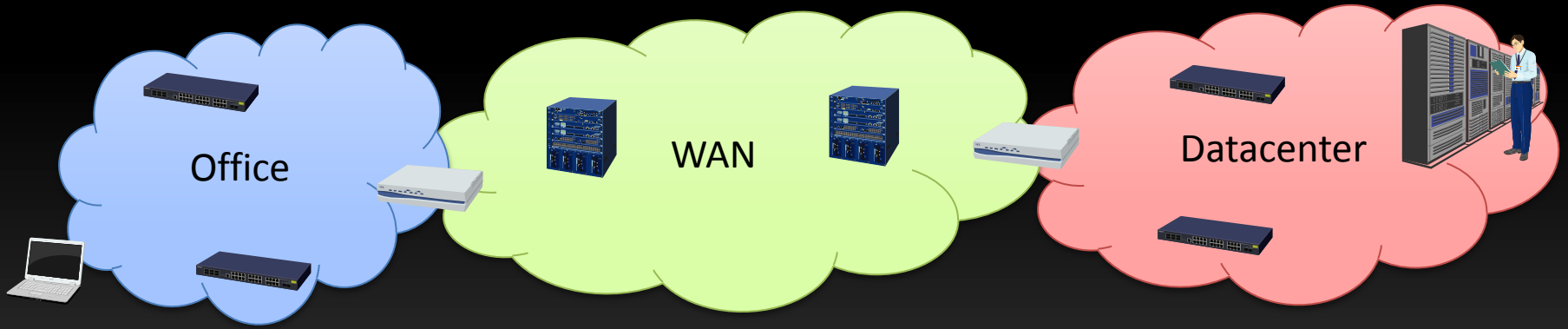


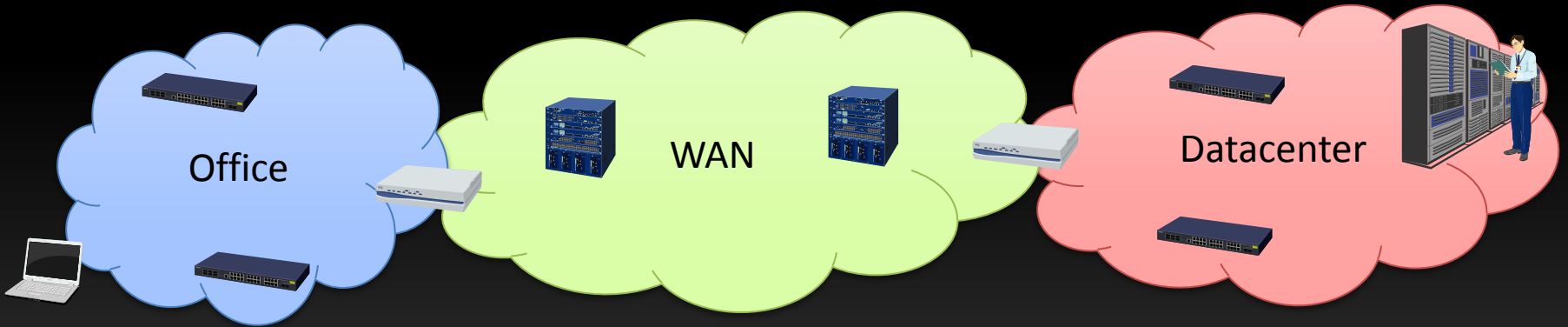
End-to-End Network Visualization and Control





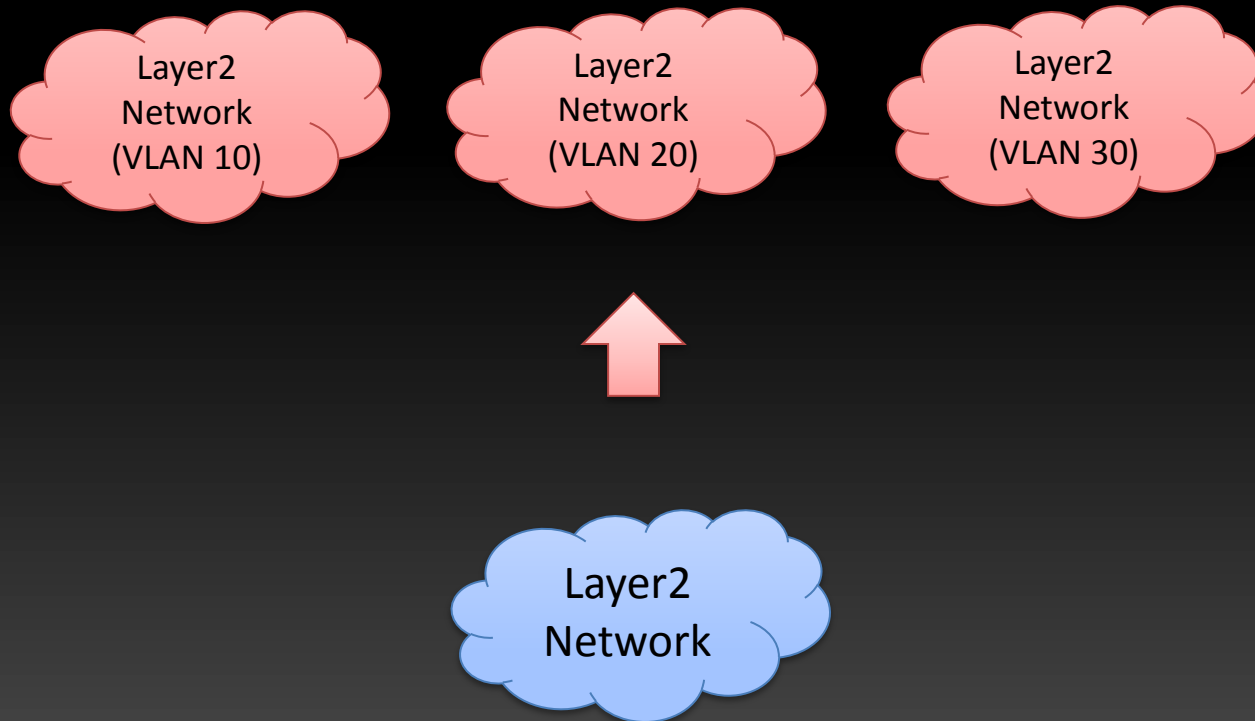




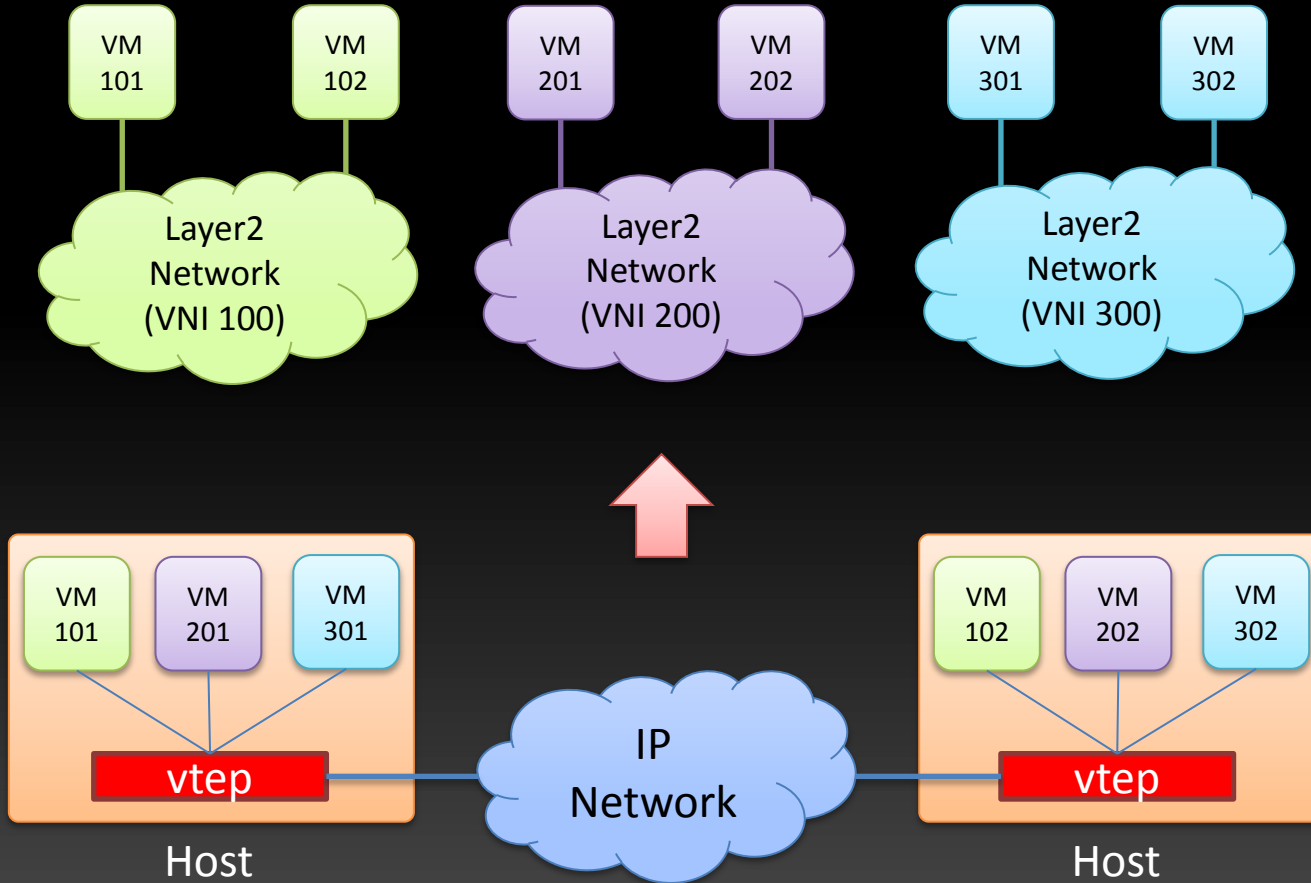


Network Virtualization

VLAN



VXLAN



Multi-Layer

Multi-Vendor  Virtualization

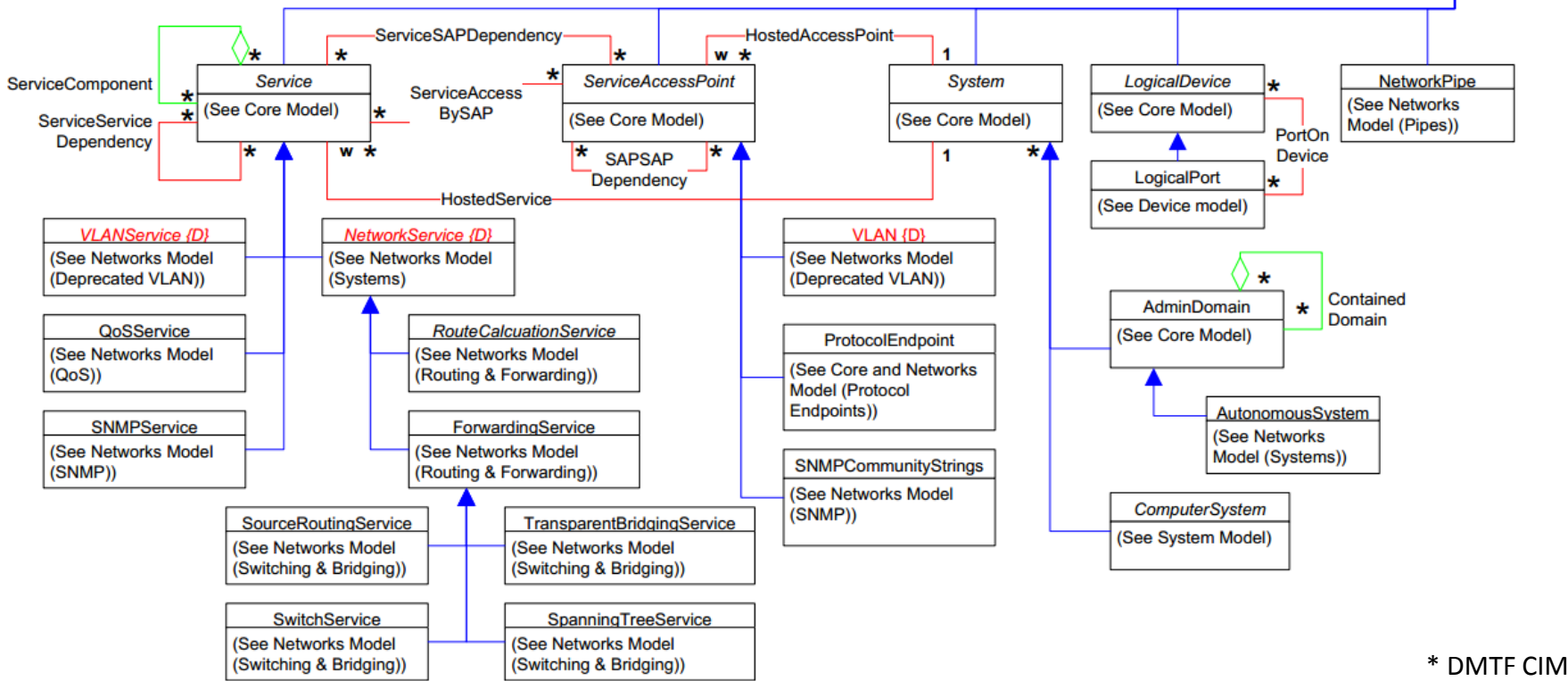
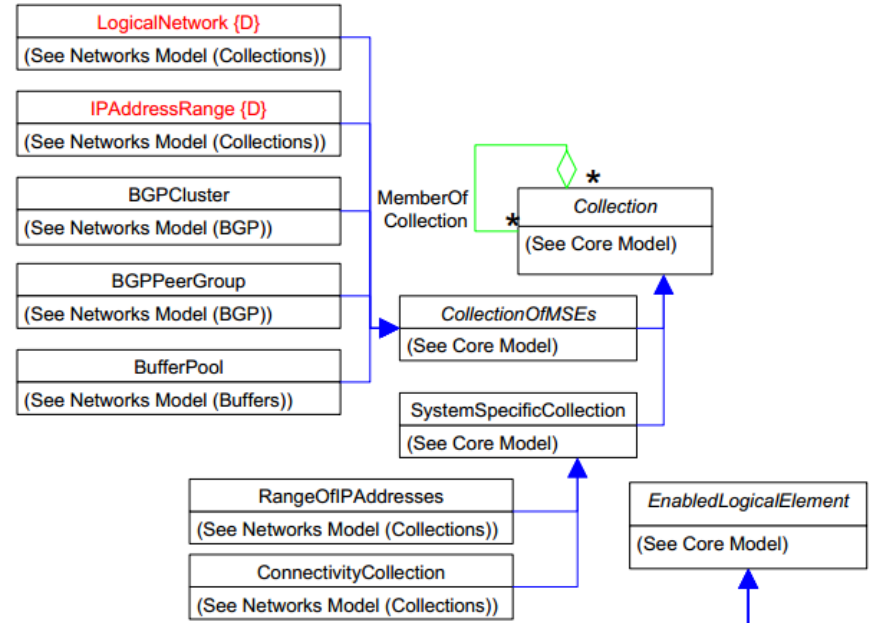
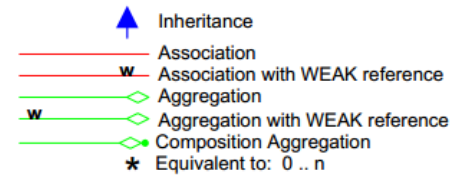
Multi-Domain

Relationship

Title : Network Specification Version 2.41.0
 Author : DMTF Networks Working Group
 Updated : 16 May 2014

Page 1,2 : Overview
 Page 3 : Network Systems
 Page 4 : Network Collections
 Page 5,6,7,8 : Protocol Endpoints (1,2,3,4)
 Page 9 : Routing and Forwarding
 Page 10 : Routes
 Page 11 : Pipes
 Page 12 : Filtering and Filter Entries
 Page 13 : Buffer Pools (Network Resources)
 Page 14 : SNMP
 Page 15 : OSPF
 Page 16,17 : BGP (1,2)
 Page 18,19 : Switching and Bridging (1,2)
 Page 20 : QoS
 Page 21,22 : QoS Conditioning Services (1,2)
 Page 23 : IPsec
 Page 24 : VLAN
 Page 25,26 : MPLS(1,2)
 Page 27 : Network Policy

Page 28 : Topology
 Page 29 : 802.3ad Link Aggregation
 Page 30 : NAT
 Page 31,32 : iSCSI (1,2)
 Page 33 : Channel Extender
 Page 34 : Redirection
 Page 35 : MediaRedirection
 Page 36 : ConfigurationServices
 Page 37,38,39 : Dependency (1,2,3)
 Page 40 : Hosted Dependency Hierarchy
 Page 41 : Association Hierarchy
 Page 42,43 : Aggregation Hierarchy (1,2)
 Page 44 : MemberOfCollection
 Page 45 : Deprecated VLAN
 Page 46,47 : Deprecated Associations (1,2)



A Mass of Specifications makes applications

→ Complicated, and

→ Non-Reuseable.



An SDN Framework for Heterogeneous Networks

ODENOS



Abstraction

Abstract Model



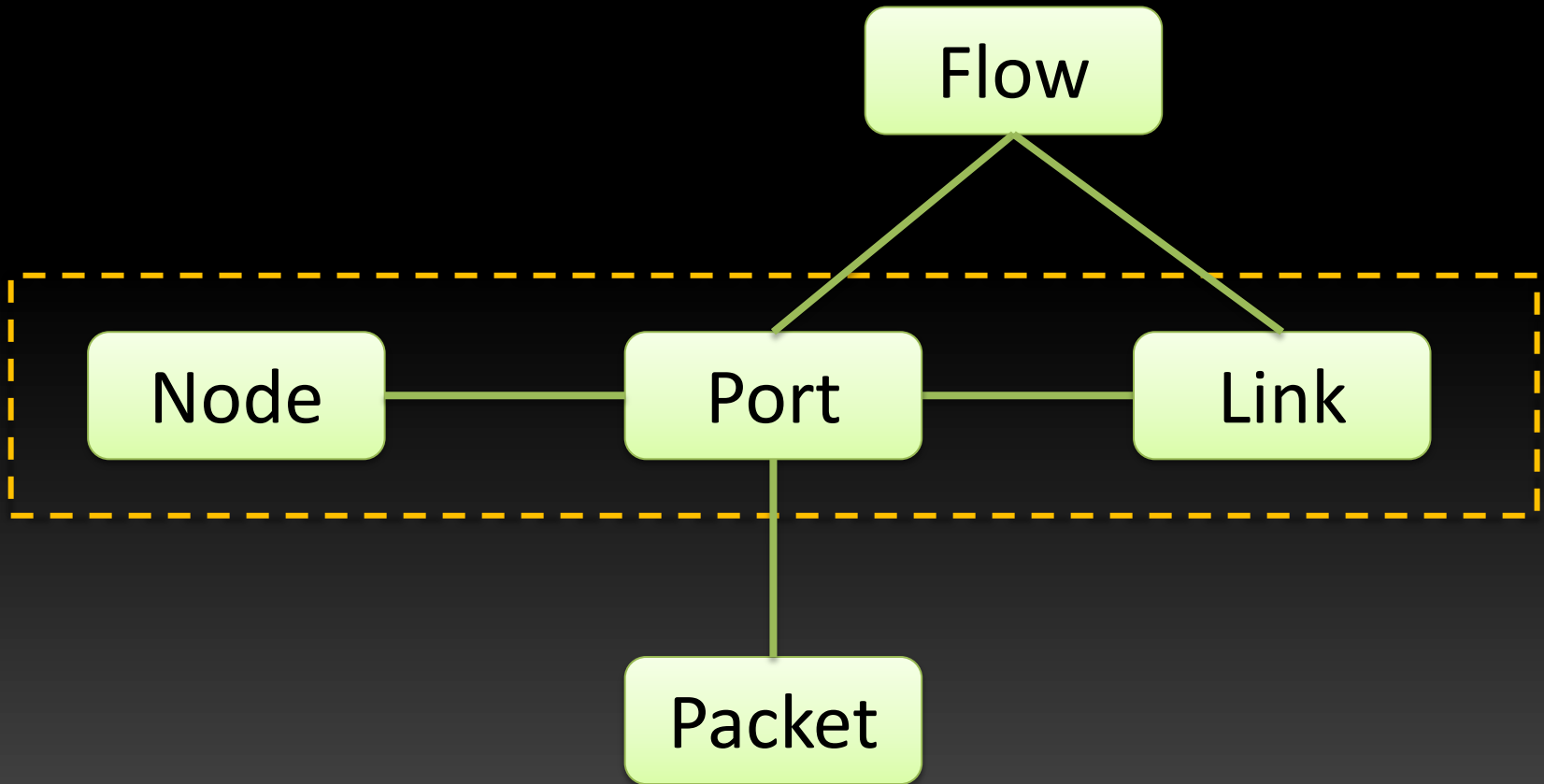
OpenFlow

Overlay

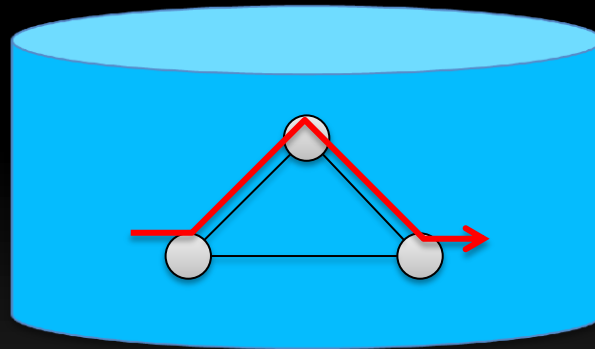
Packet
Transport

Optical
Transport

Network Model

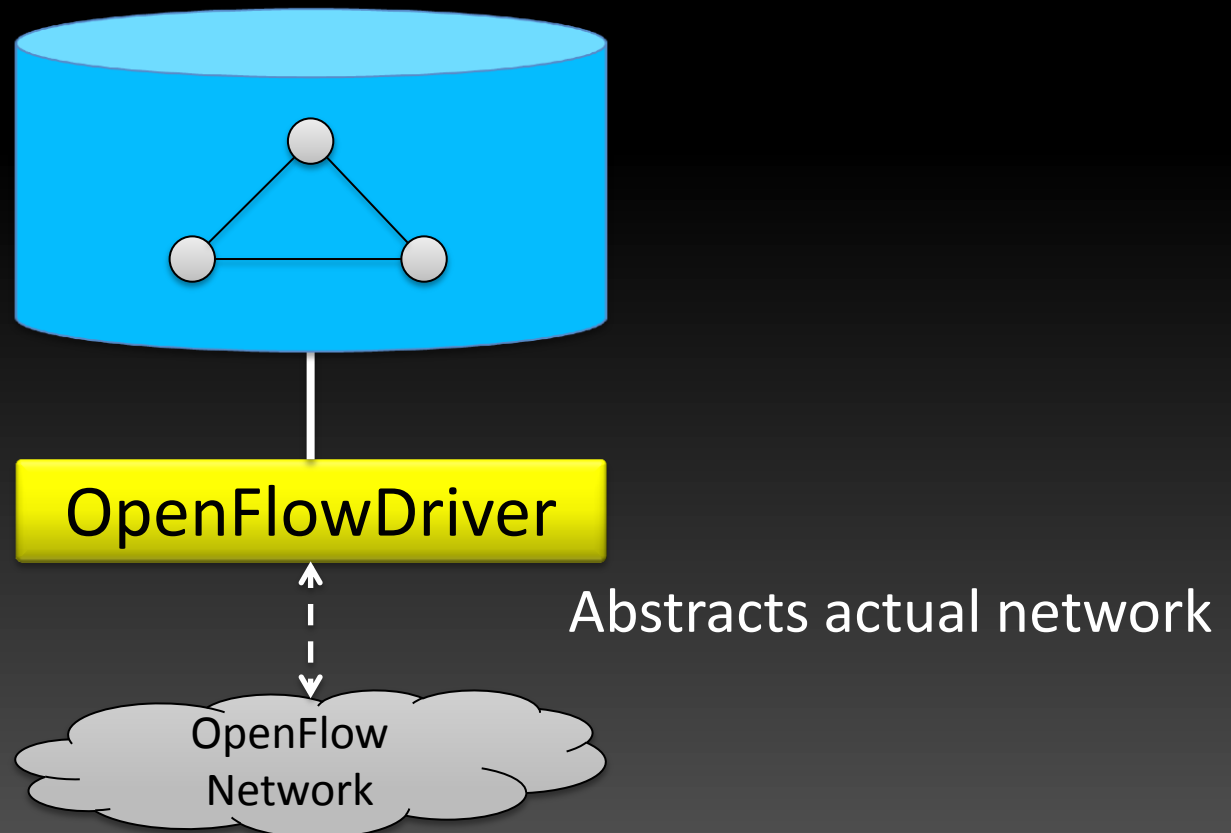


Network component

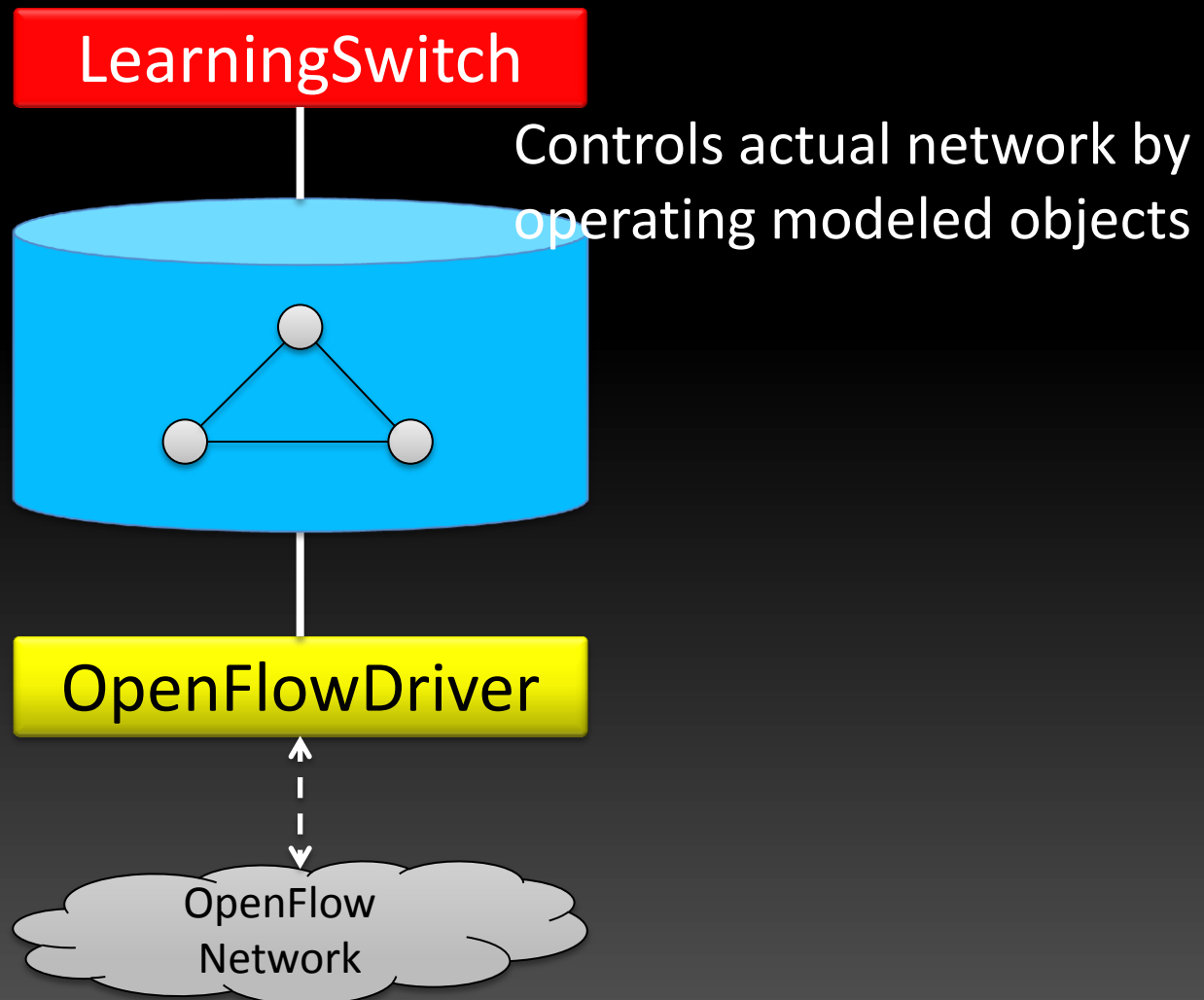


Stores modeled objects

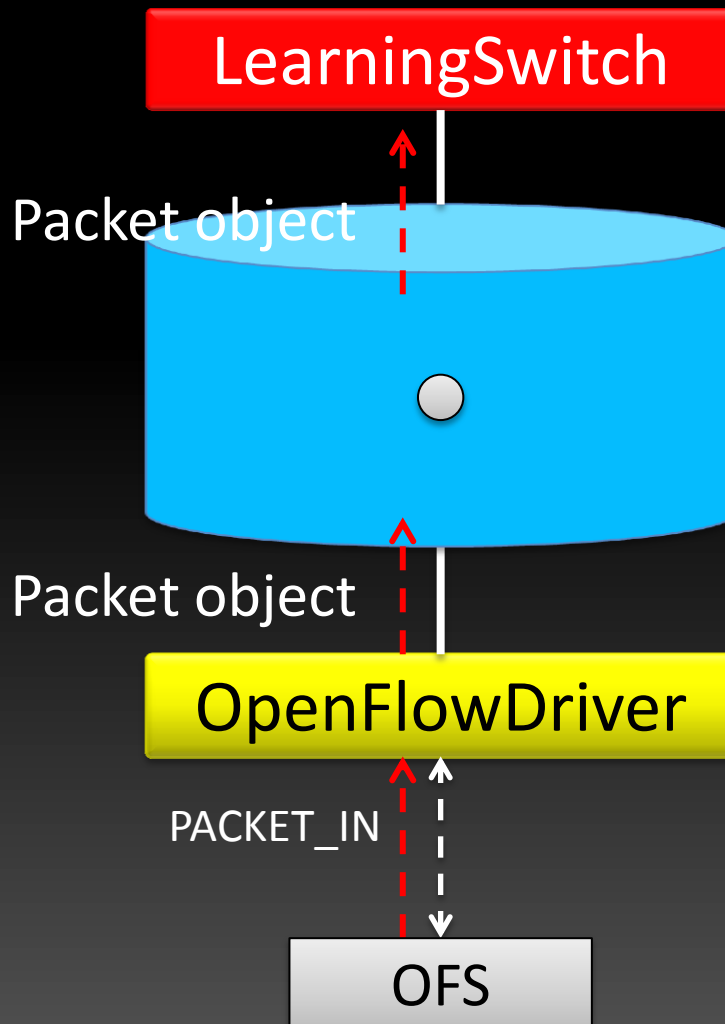
Driver component



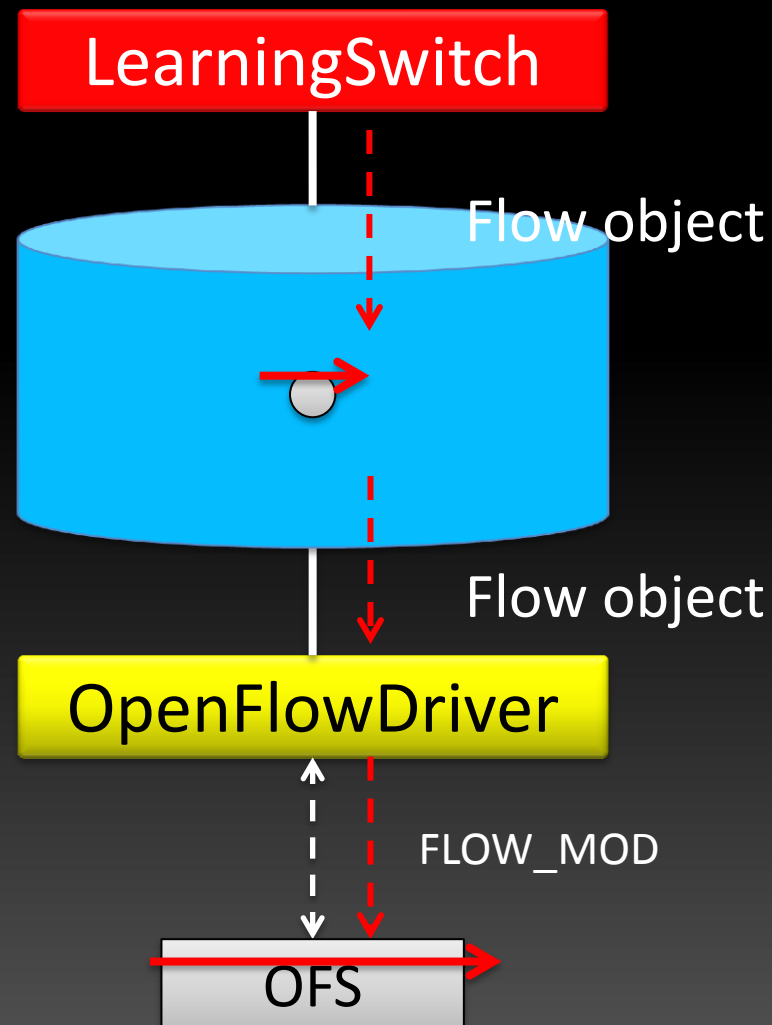
Application component



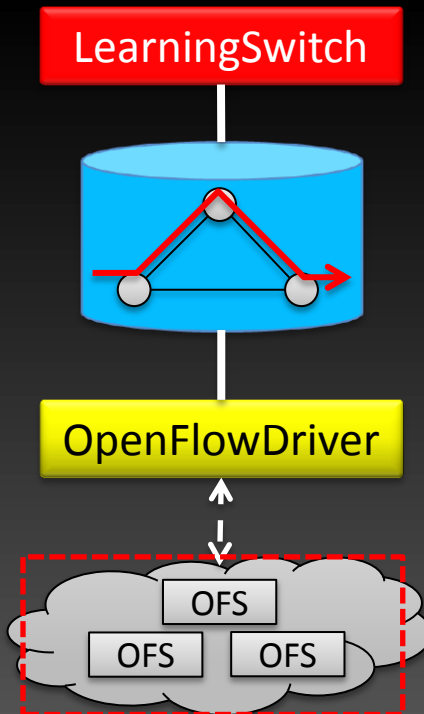
LearningSwitch Application



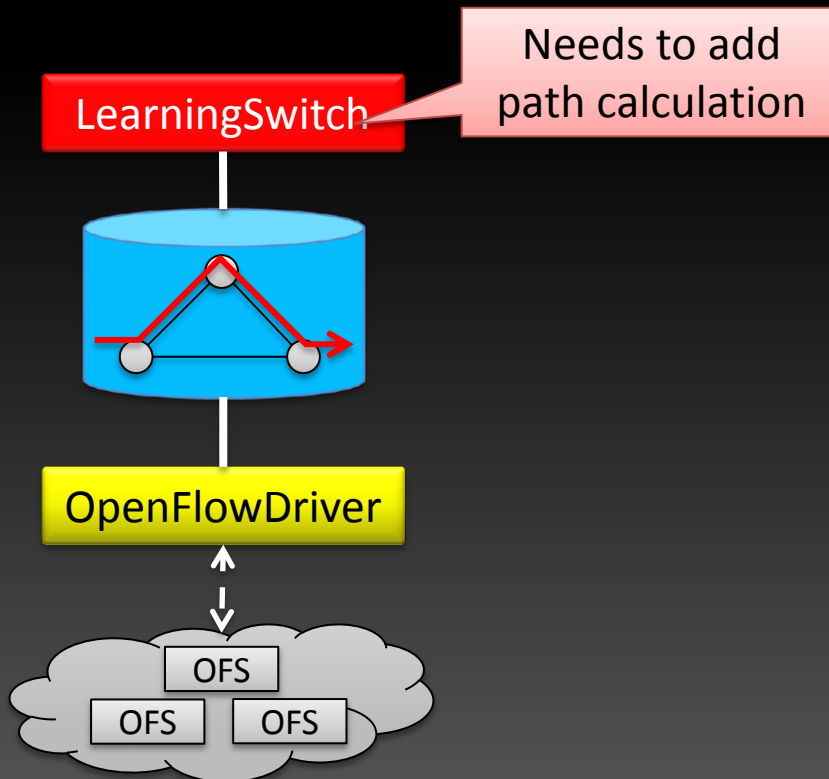
Controlling Single OpenFlow switch



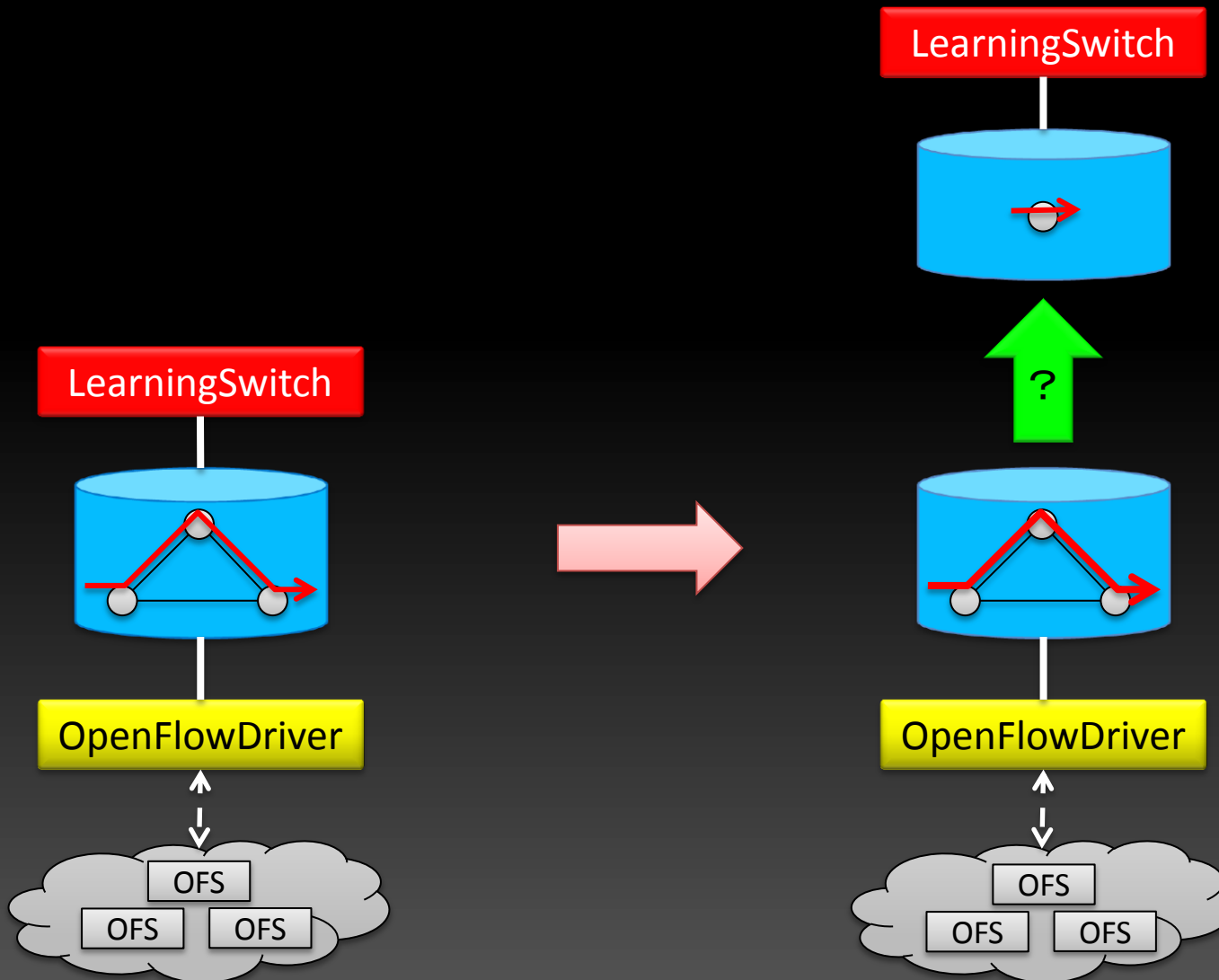
Controlling Multiple OpenFlow switches



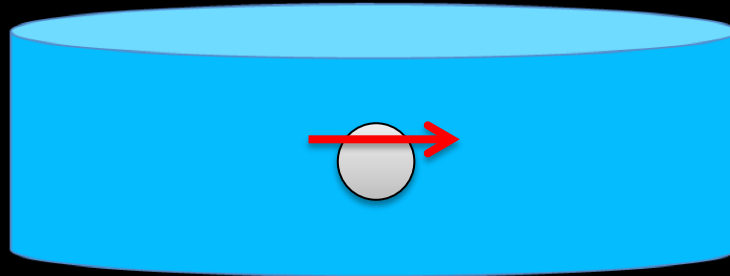
Controlling Multiple OpenFlow switches



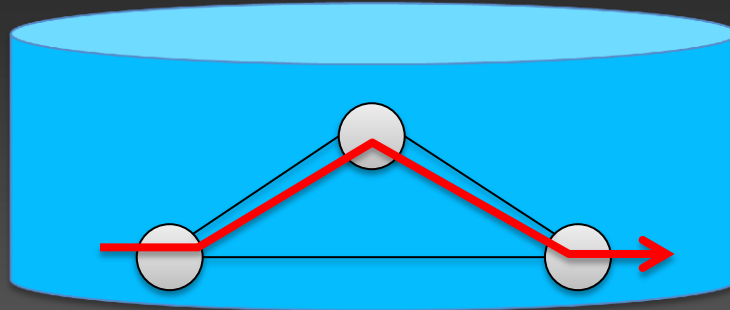
Converting modeled objects



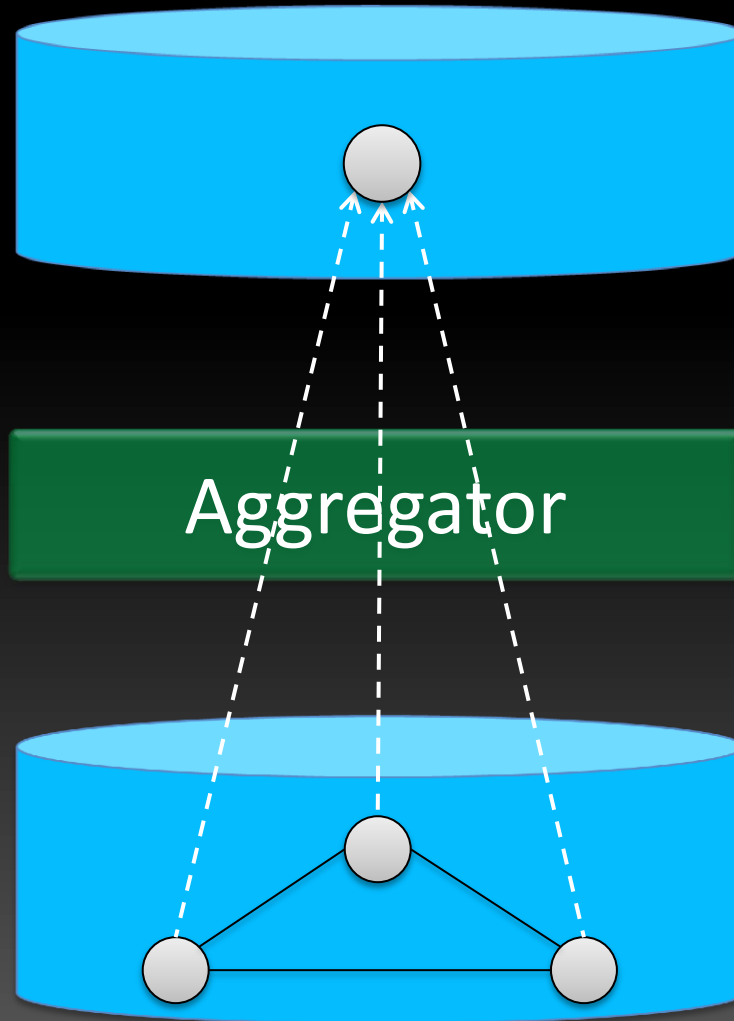
How to convert?



Aggregator

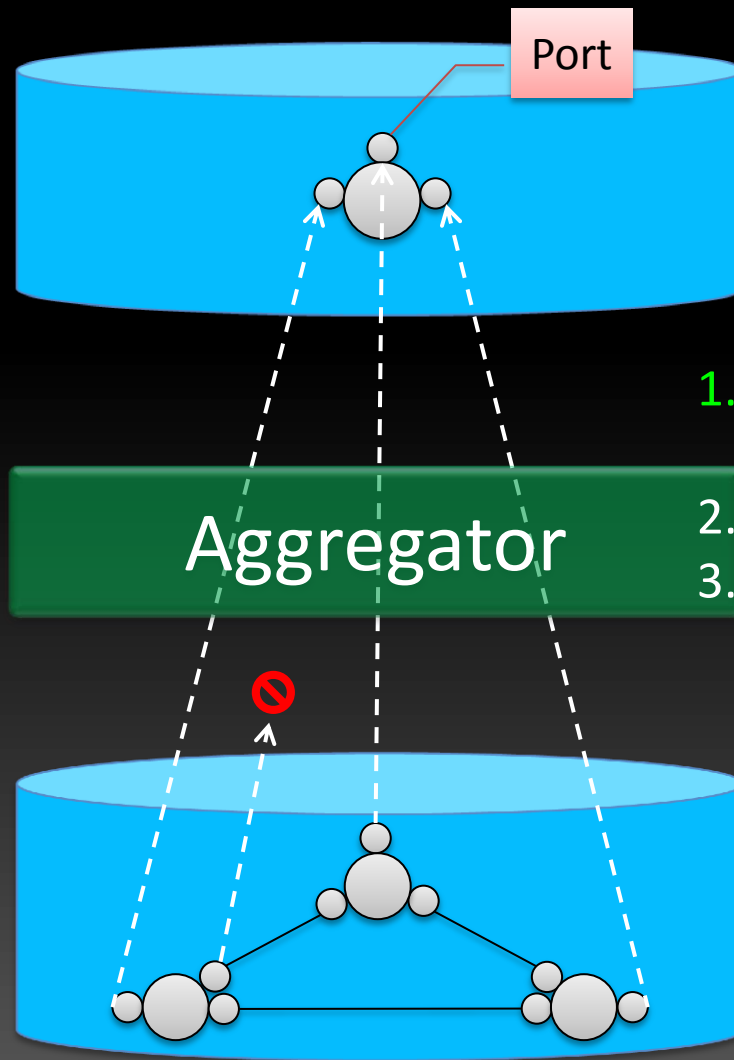


Topology



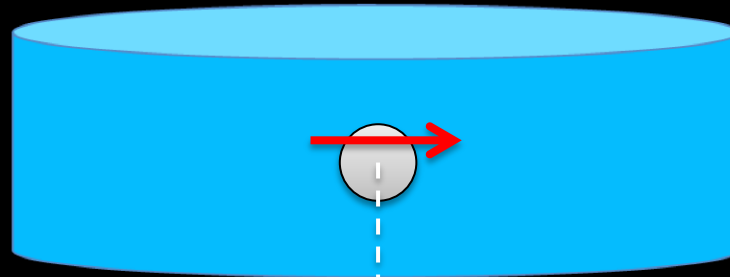
1. **Aggregates** multiple nodes to single node.

Topology

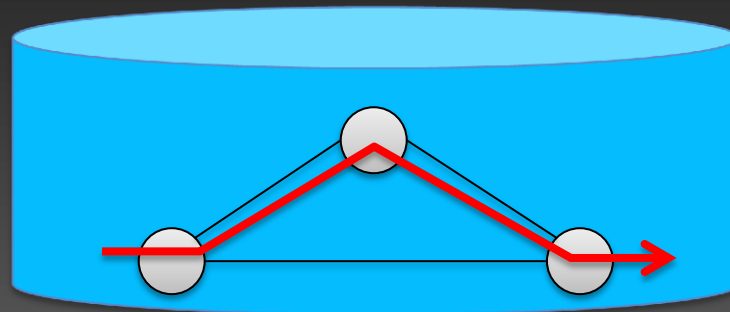


1. **Aggregates** multiple nodes to single node.
2. Copies outer ports
3. Discards inner ports

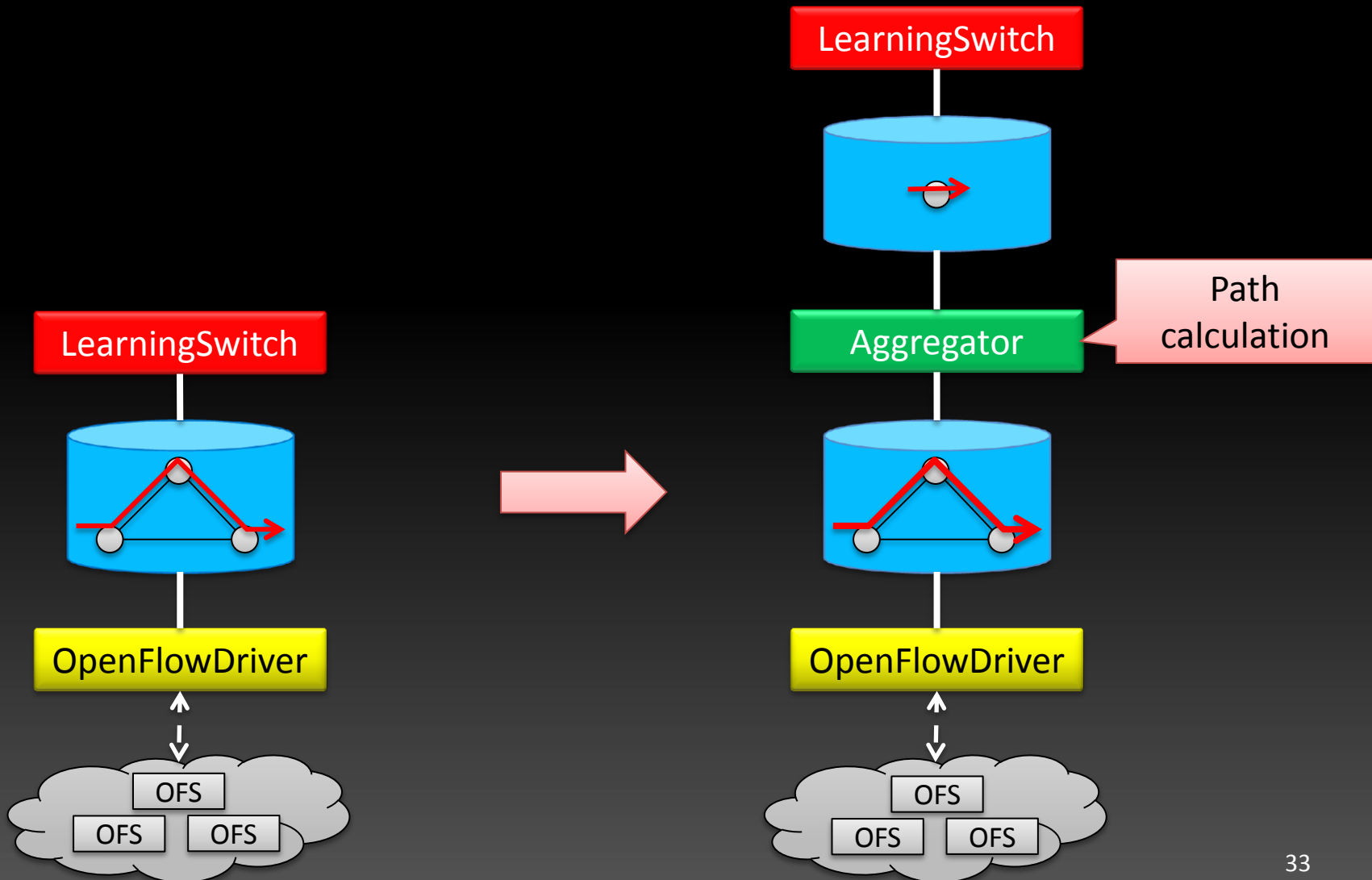
Flow



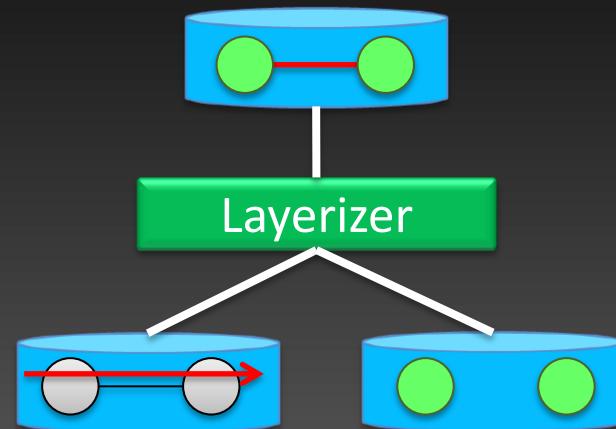
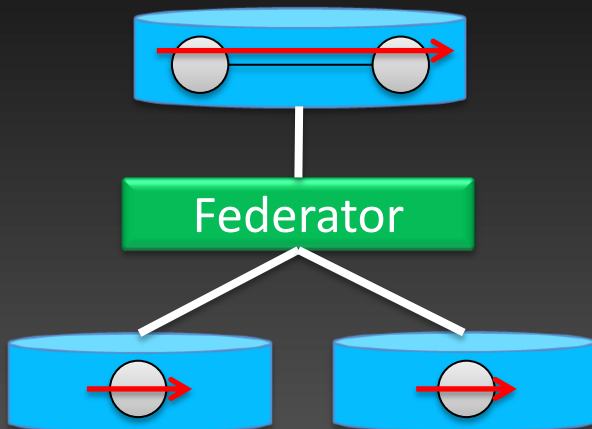
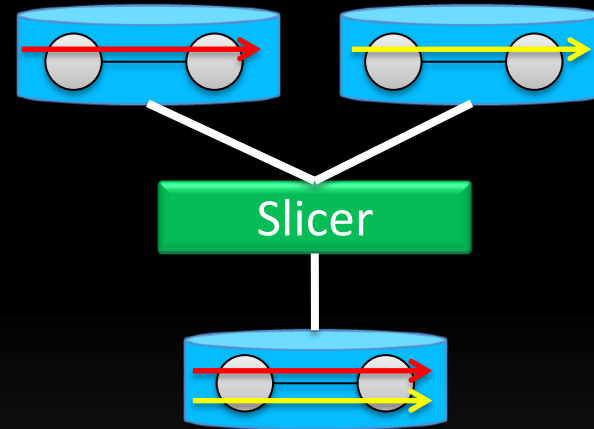
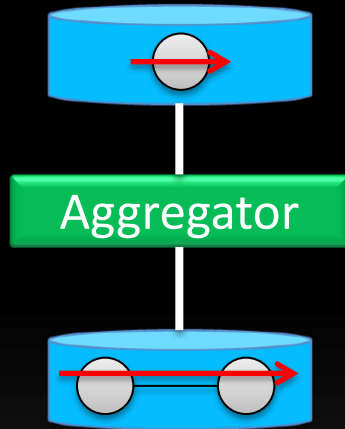
Expands the flow
by path calculation



Reuse of Applications

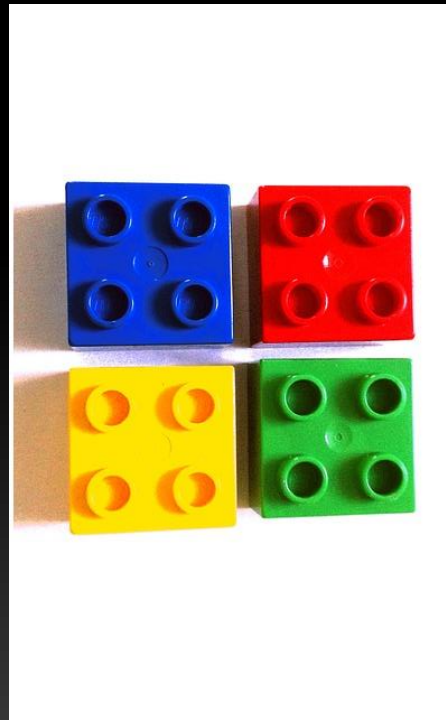


Logic Components



Network

Application

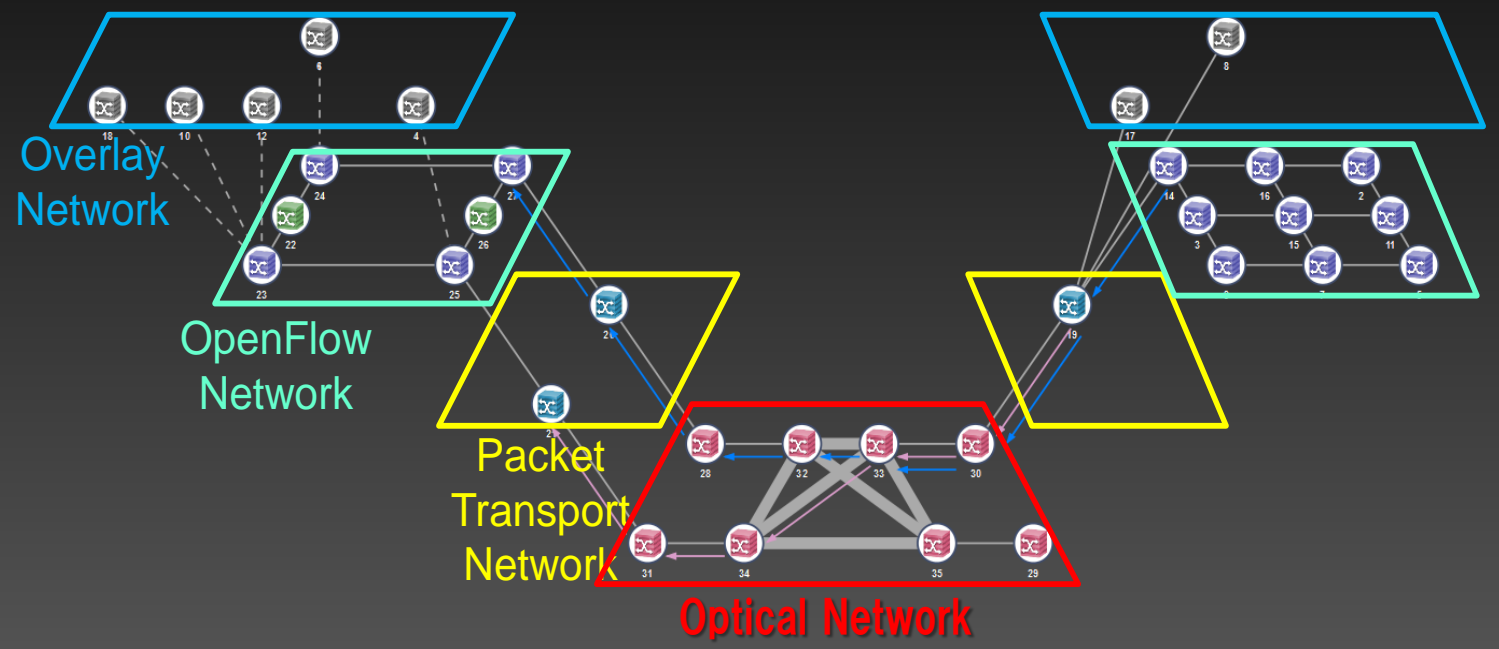


Driver

Logic

Network Visualization for Heterogeneous Networks

Auto layout
+ -



Auto Refresh Interval 10 sec.

Refresh

Node	Flow	Link	Port	Group
network012	f0003_0002_1	BasicFlow	establishing	dummy true 65535 in_port=cport01.in_node=node2_0003

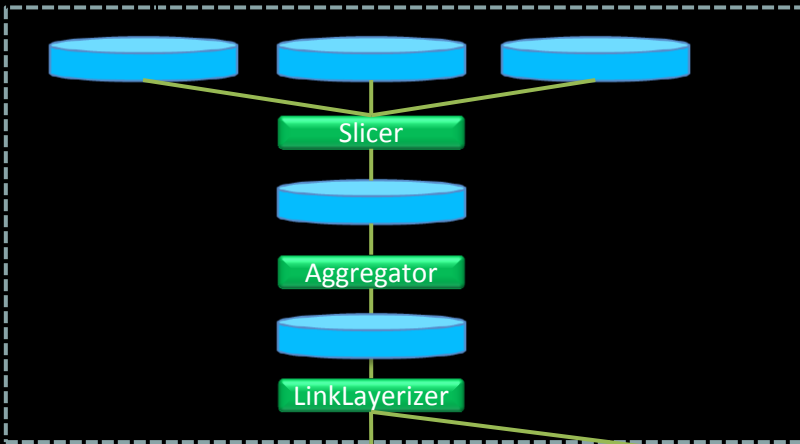
Path: 012345_0003_0002_1_l2u
b012345_0003_0002_1_u2l,b012_0004_0003_1_u2l,b01_0004_0004_2_u2l,l0004_0002_2,b01_0002_0002_4_l2u,b012_0002_0002_4_l2u,b012345_0002_0005_1_l2u

Prev 1 Next Search Save

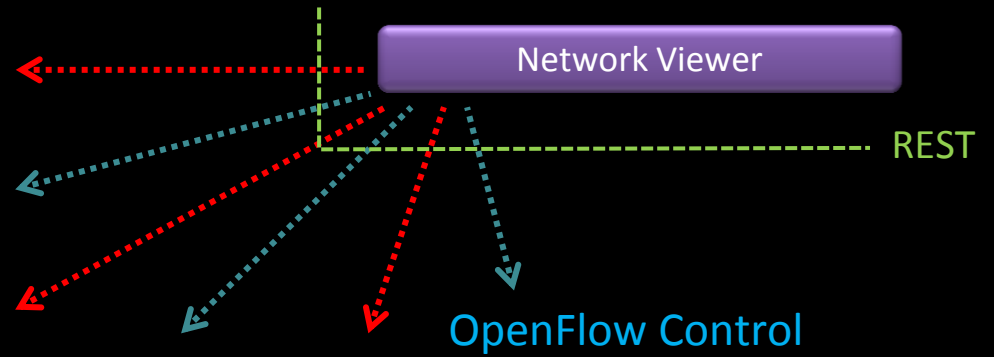
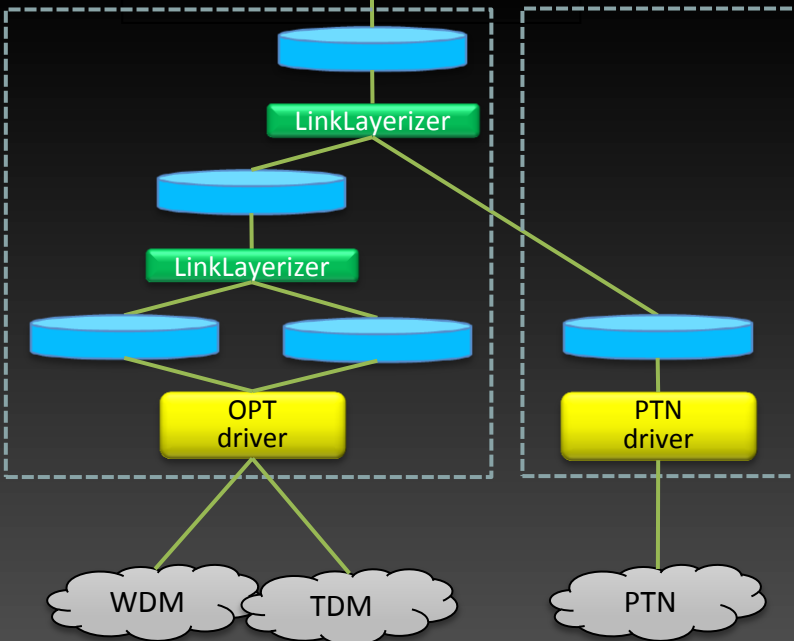
Attribute	Value

Save

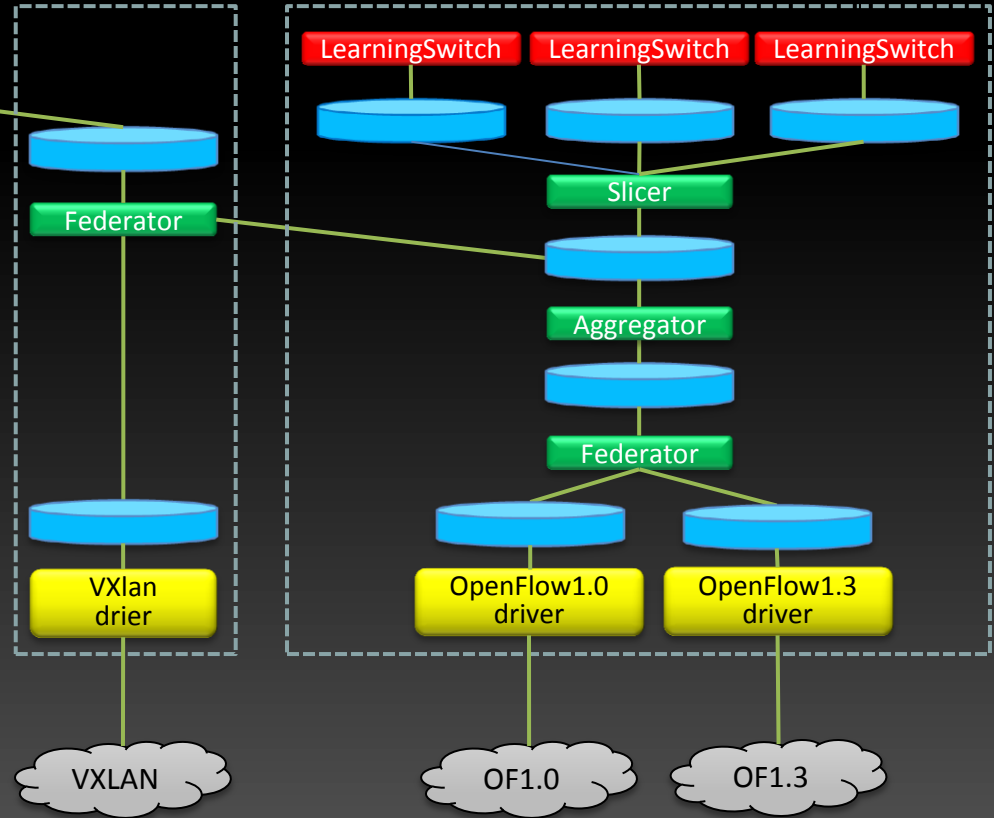
Overlay Control

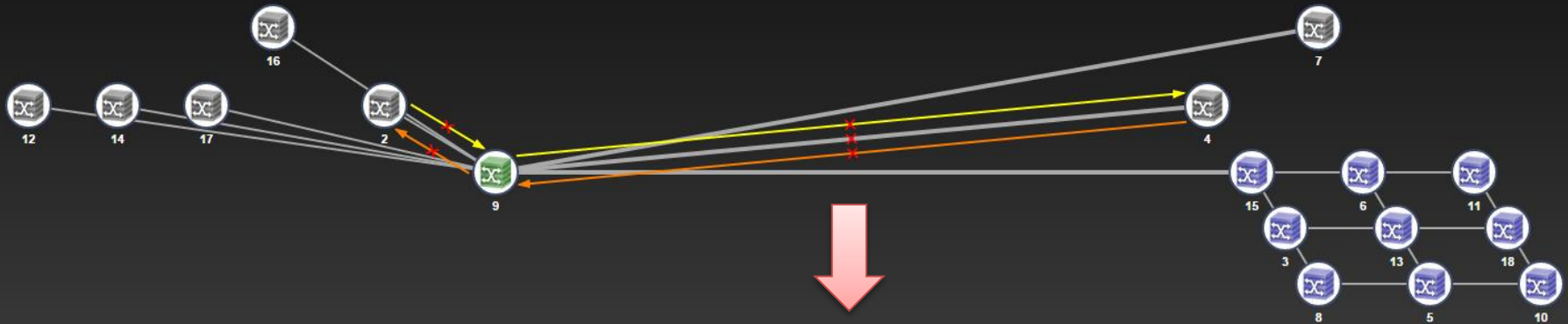


Transport Control

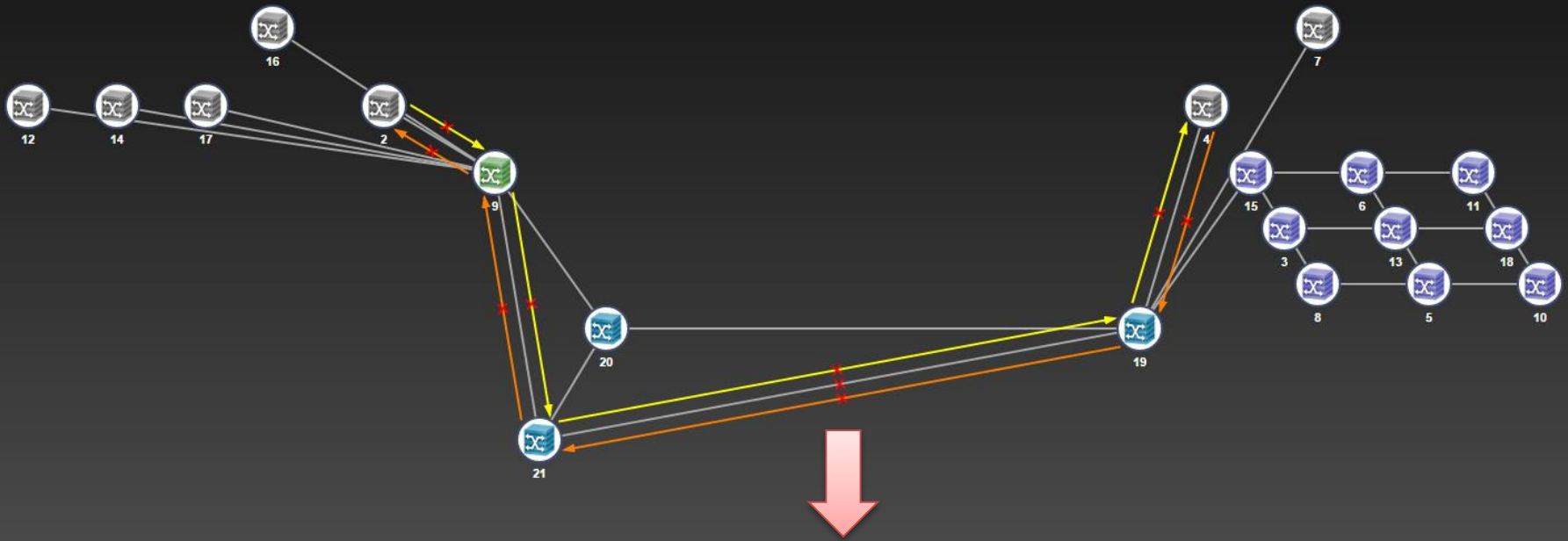


OpenFlow Control

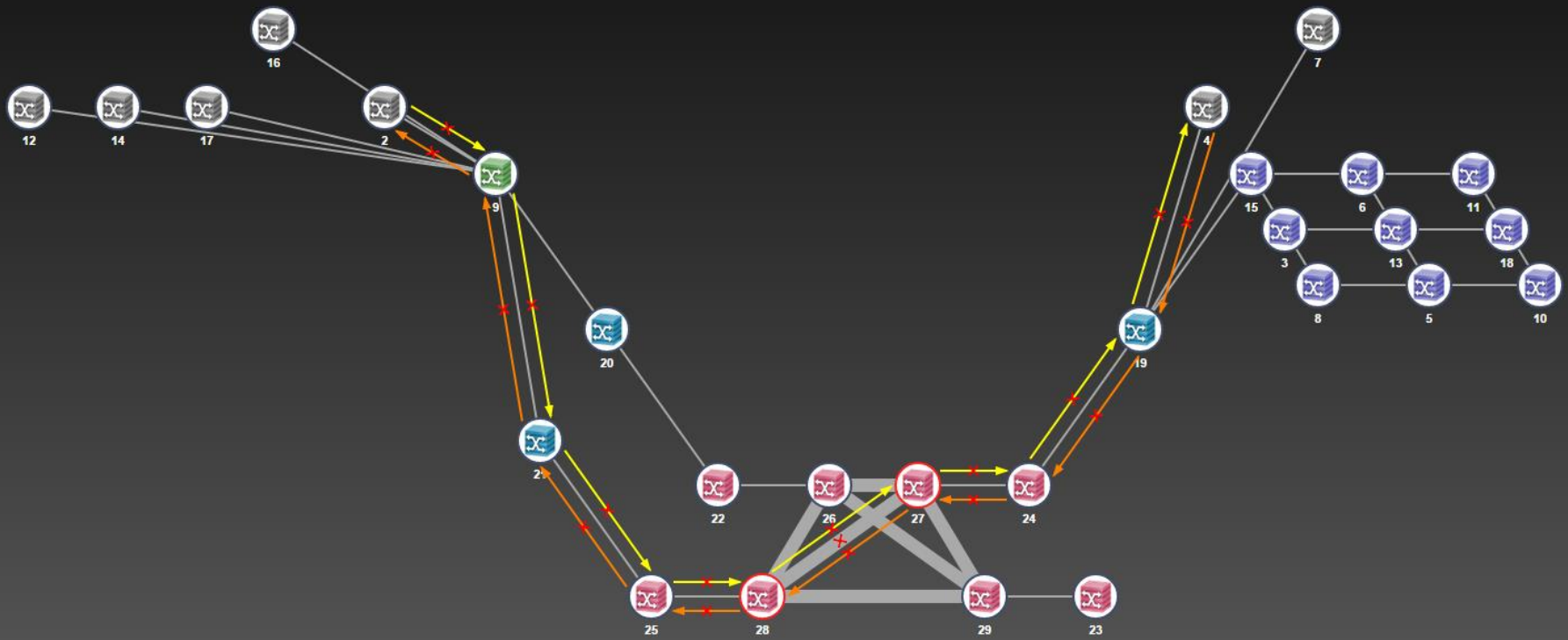




Failure of End-to-end Communication



Cause Analysis by Drilled-Down Operation



Cause Analysis by Drilled-Down Operation

Multi-layer and Multi-domain Network Control

Global Restoration

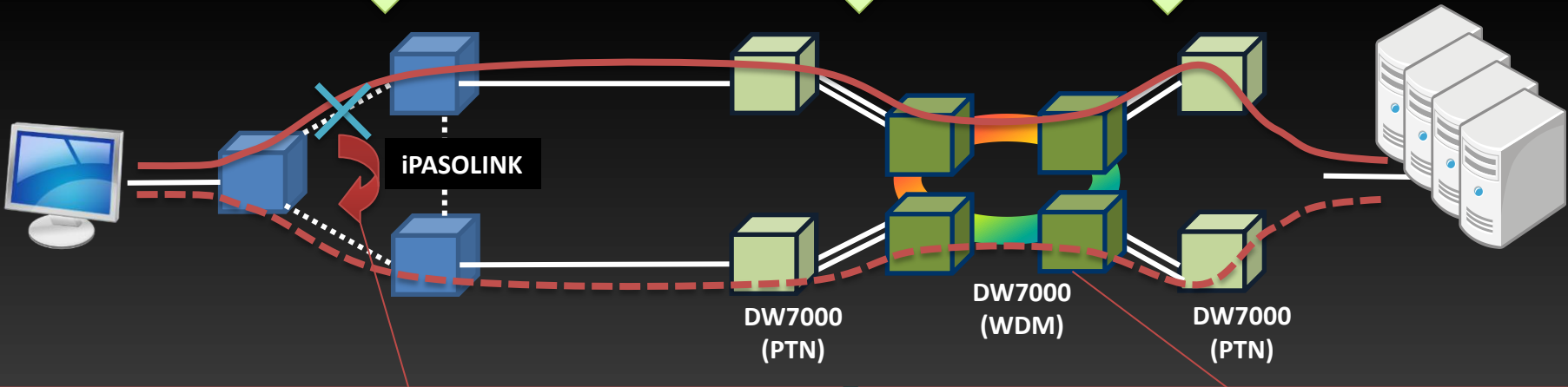
Multi-layer Path Setup

Network Orchestrator (ODENOS)

Wireless Transport Driver

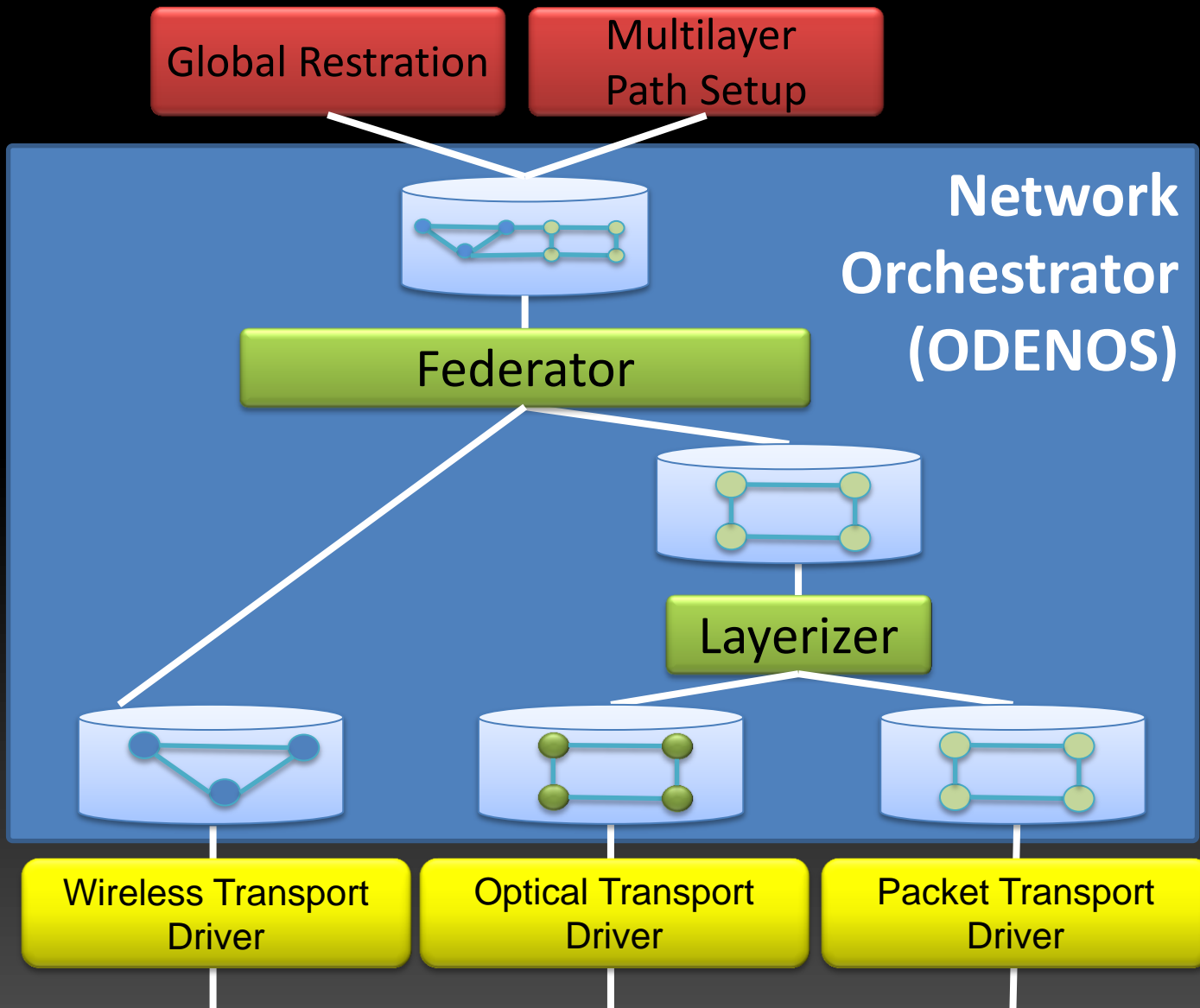
Optical Transport Driver

Packet Transport Driver



Global Restoration

Multilayer Path Calculation



Conclusion

- **ODENOS** : An SDN framework for heterogeneous network orchestration
 - Possible to build **reusable** network orchestrator **easily**
 - Flexible **end-to-end** control of **multi-layer and multi-domain** network
- **Open source software**, distributed on GitHub now!
 - Support Java and python
 - <https://github.com/o3project/odenos>

APPENDIX

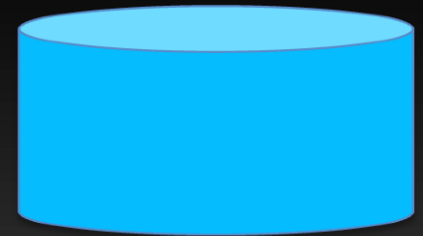
Creating Components

```
$ curl http://localhost:10080/systemmanager/components -X POST -d  
'{"type": "LearningSwitch", "id": "learning_sw1"}'
```

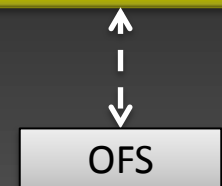
```
$ curl http://localhost:10080/systemmanager/components -X POST -d  
'{"type": "NetworkComponent", "id": "network1"}'
```

```
$ curl http://localhost:10080/systemmanager/components -X POST -d  
'{"type": "OpenFlowDriver", "id": "driver1"}'
```

LearningSwitch



OpenFlowDriver



Connecting Components

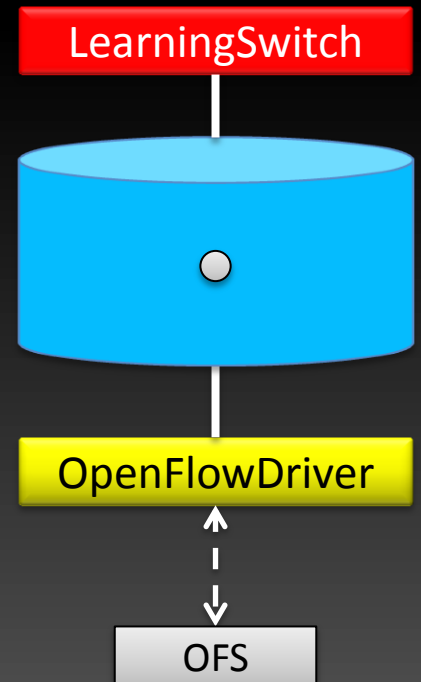
```
$ curl http://localhost:10080/systemmanager/components -X POST -d  
{ "type": "LearningSwitch", "id": "learning_sw1" }
```

```
$ curl http://localhost:10080/systemmanager/components -X POST -d  
{ "type": "NetworkComponent", "id": "network1" }
```

```
$ curl http://localhost:10080/systemmanager/components -X POST -d  
{ "type": "OpenFlowDriver", "id": "driver1" }
```

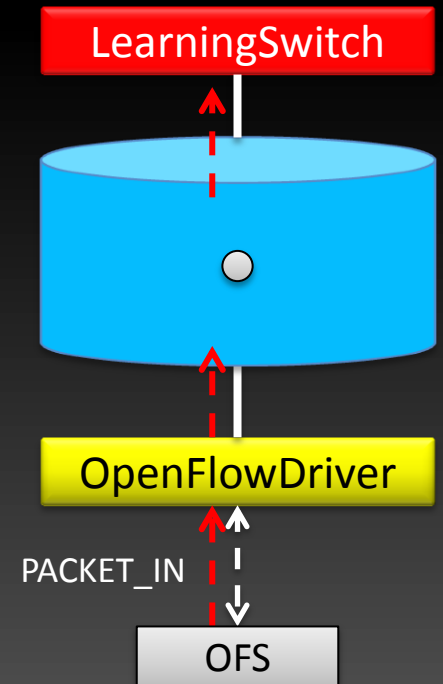
```
$ curl http://localhost:10080/systemmanager/connections -X POST -d  
{ "connection_type": "original",  
  "source_id": "learning_sw1", "destination_id": "network1" }
```

```
$ curl http://localhost:10080/systemmanager/connections -X POST -d  
{ "connection_type": "original",  
  "source_id": "driver1", "destination_id": "network1" }
```

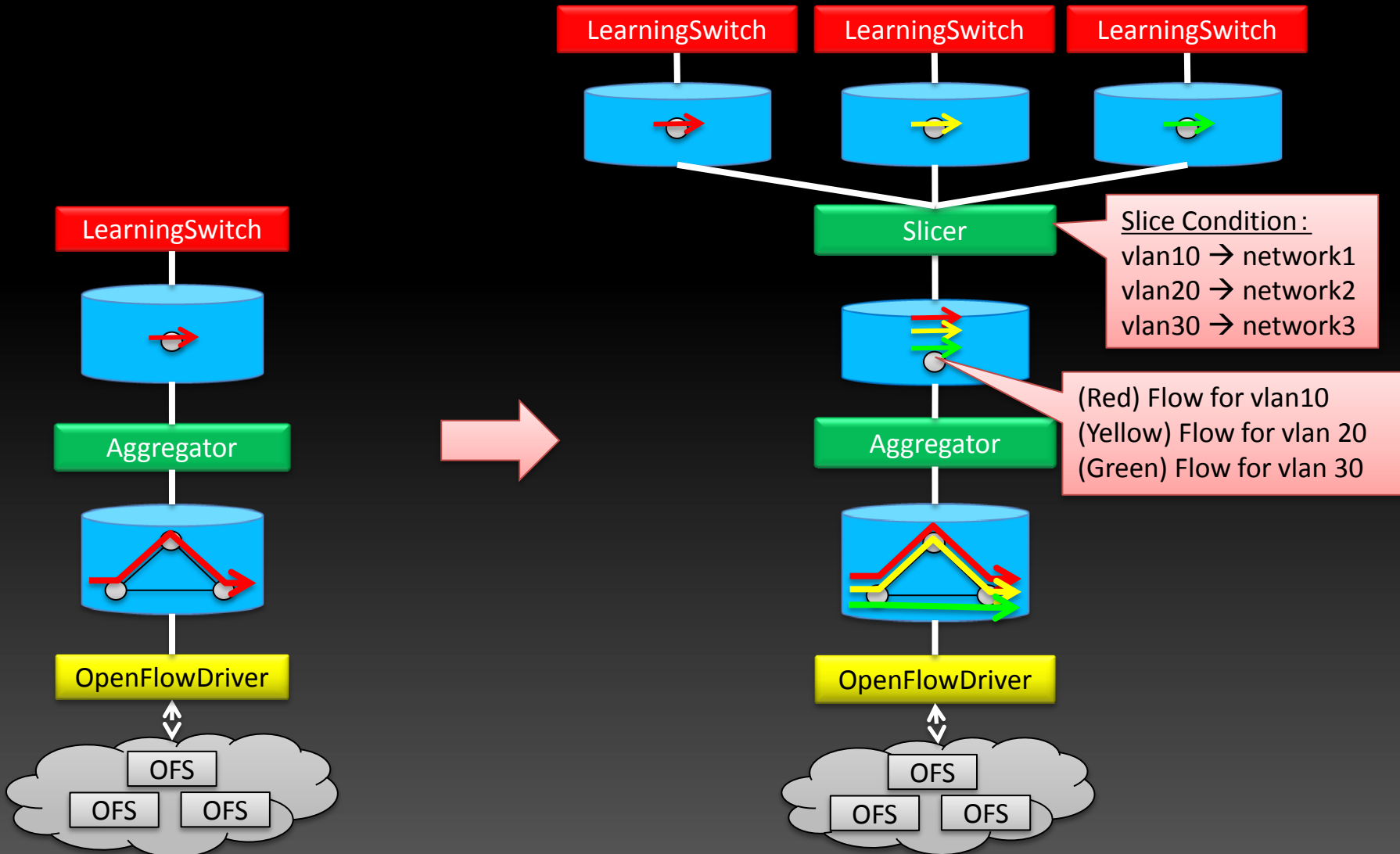


Implementation of LearningSwitch Application

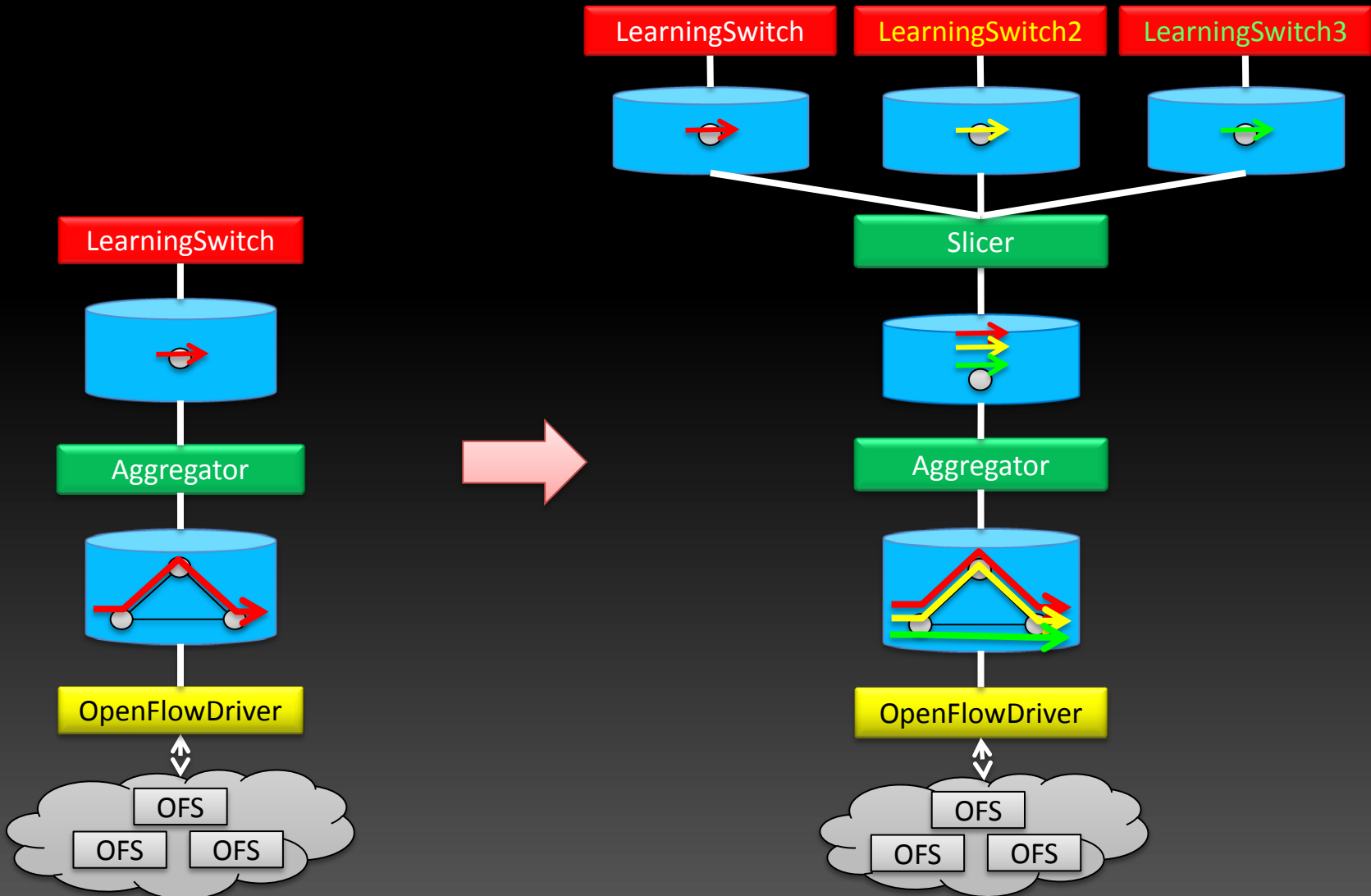
```
class LearningSwitch extends Logic {  
    ...  
    network.addInPacketAddedListener(  
        new InPacketAddedListener() {  
            ...  
            public void processEvent(final InPacketAdded msg) {  
                // Receives a packet  
                InPacket pkt = network.getPacket(msg.id());  
  
                // MAC Learning  
                inPort = fdb.put(pkt.srcMacAddr(), pkt.inPort());  
  
                // If the MAC has been learned, creates Flow  
                outPort = fdb.get(pkt.dstMacAddr());  
                if (outPort != NULL) {  
                    network.putFlow(  
                        Util.createL2Flow(pkt, outPort));  
                }  
  
                // Sends a packet  
                network.postOutPacket(  
                    Util.createOutPacket(pkt, outPort));  
            }  
        }  
    );  
    ...  
}
```



Slicer : A Logic Component



Slicer : A Logic Component

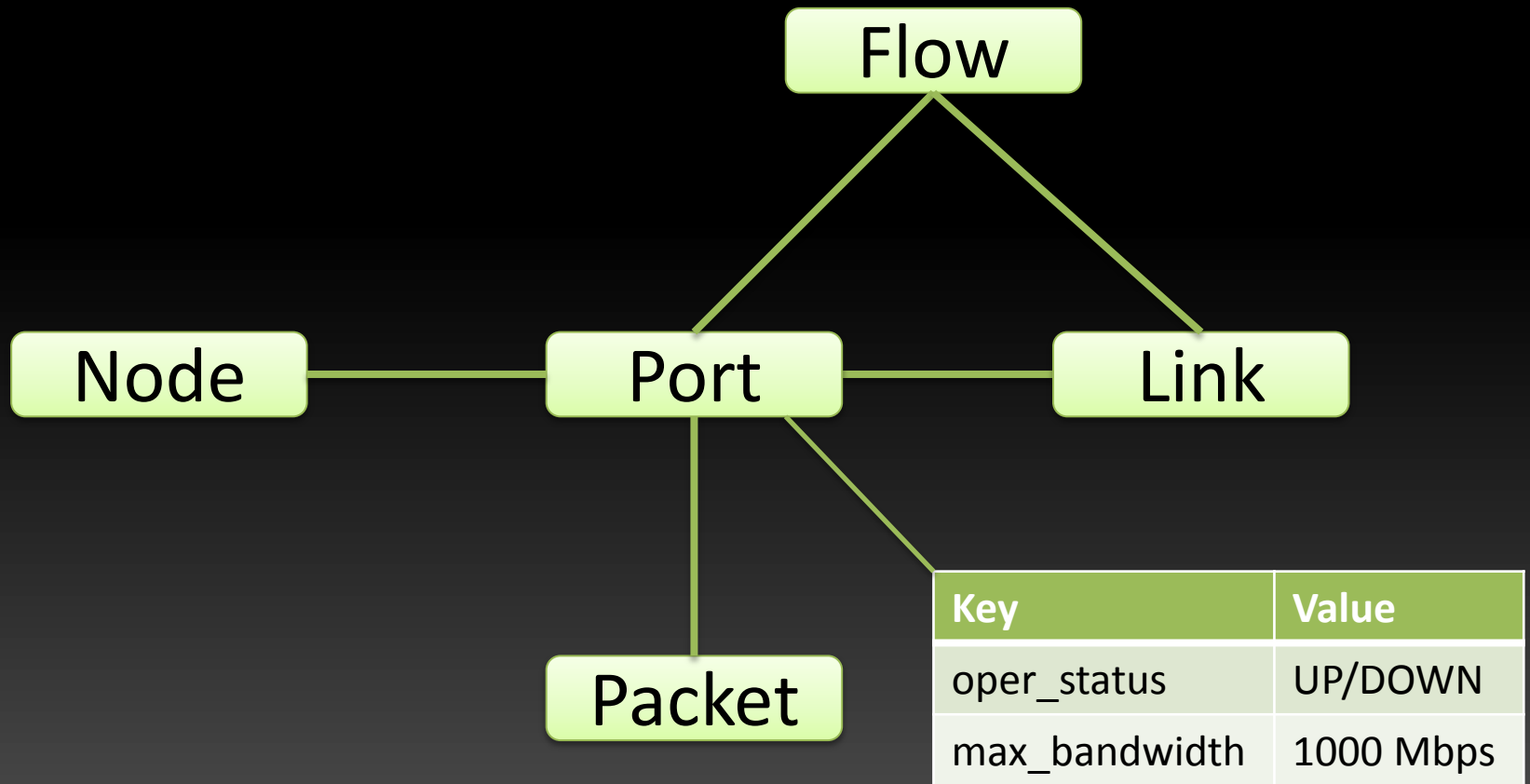




Q. Can I expand models or operators?

A. Yes, it's possible.

Network Model



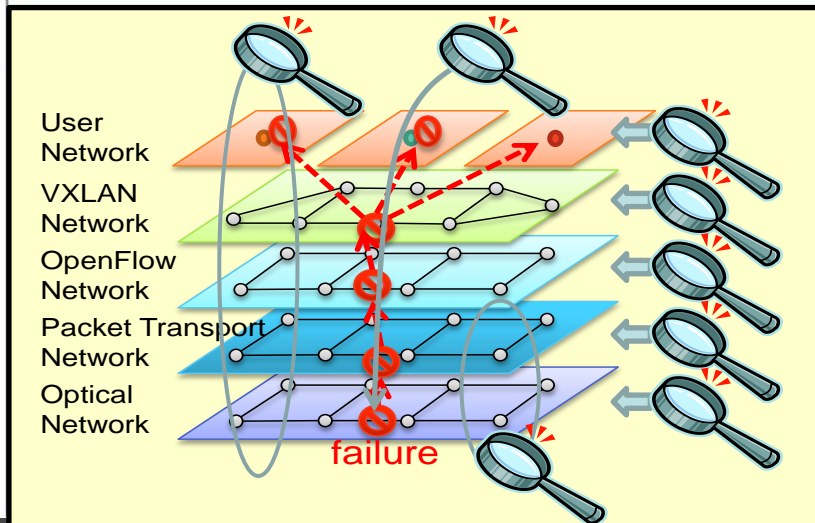
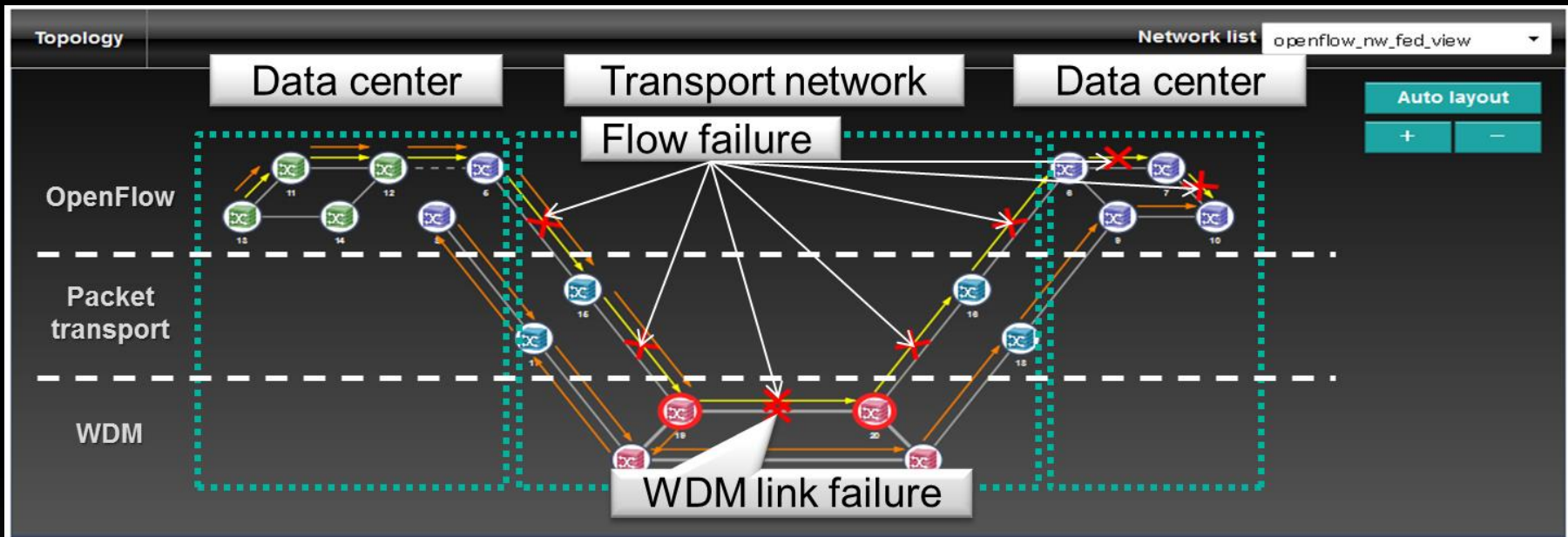
```
graph BT; AA[Advanced Aggregator] -- Inheritance --> A[Aggregator];
```

Aggregator

Inheritance

Advanced Aggregator

Overrides necessary
methods



- Multi-layer topology visualization from logical network instances
- Inter-layer correlation mapping through operators
- Trouble shooting, failure analysis, etc.