ON.LAB

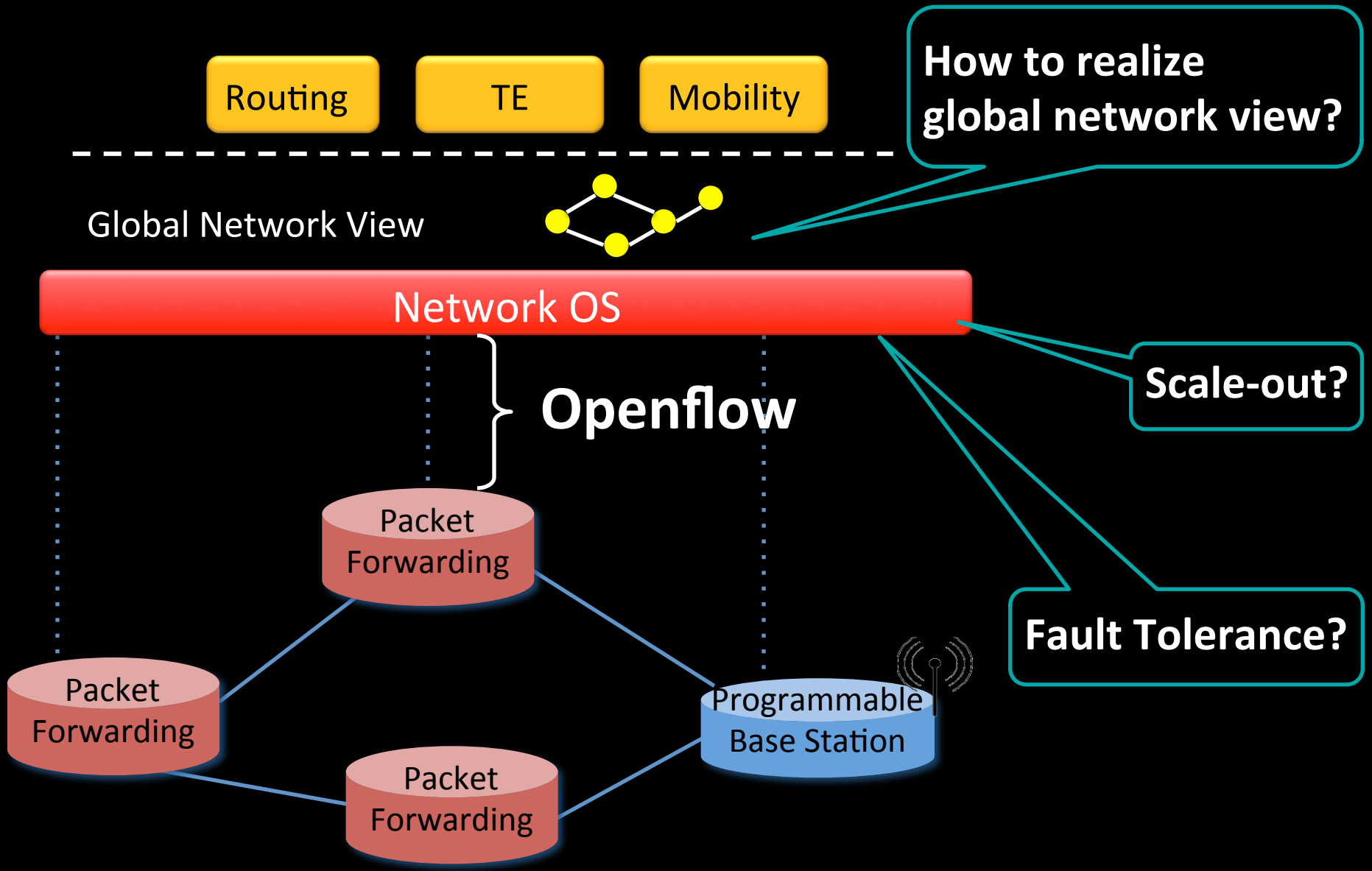
ONOS

Open Network Operating System

An Experimental Open-Source Distributed SDN OS

Umesh Krishnaswamy, Pankaj Berde, Jonathan Hart, Masayoshi Kobayashi, Pavlin Radoslavov, Tim Lindberg, Rachel Sverdlov, Suibin Zhang, William Snow, Guru Parulkar

Logically Centralized NOS – Key Questions



Prior Work

ONIX

- Distributed control platform for large-scale networks
- Focus on reliability, scalability, and generality
- State distribution primitives, global network view, ONIX API

Other Work

- Helios (NEC), Midonet (Midokura), Hyperflow, Maestro, Kandoo
- NOX, POX, Beacon, Floodlight, Trema controllers

Community needs an open source distributed SDN OS

ONOS High Level Architecture

Network Graph
Eventually consistent

Titan Graph DB

Cassandra In-Memory DHT

Distributed Registry
Strongly Consistent

Zookeeper

Instance 1

OF Controller

Floodlight

Instance 2

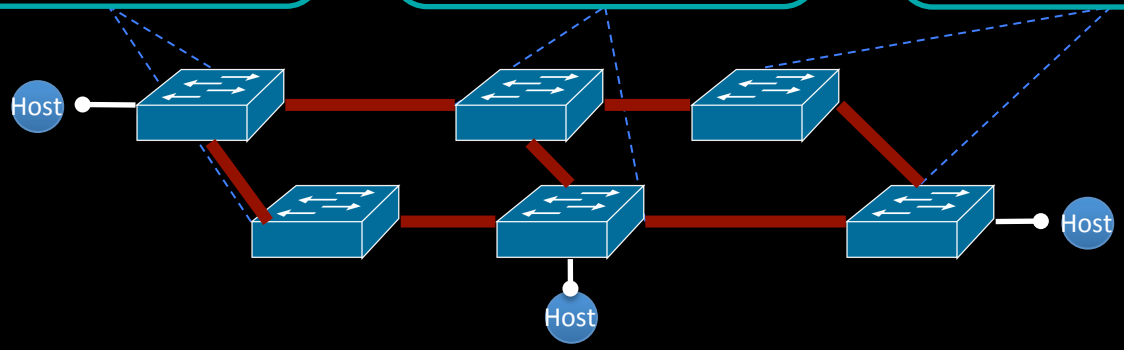
OF Controller

Floodlight

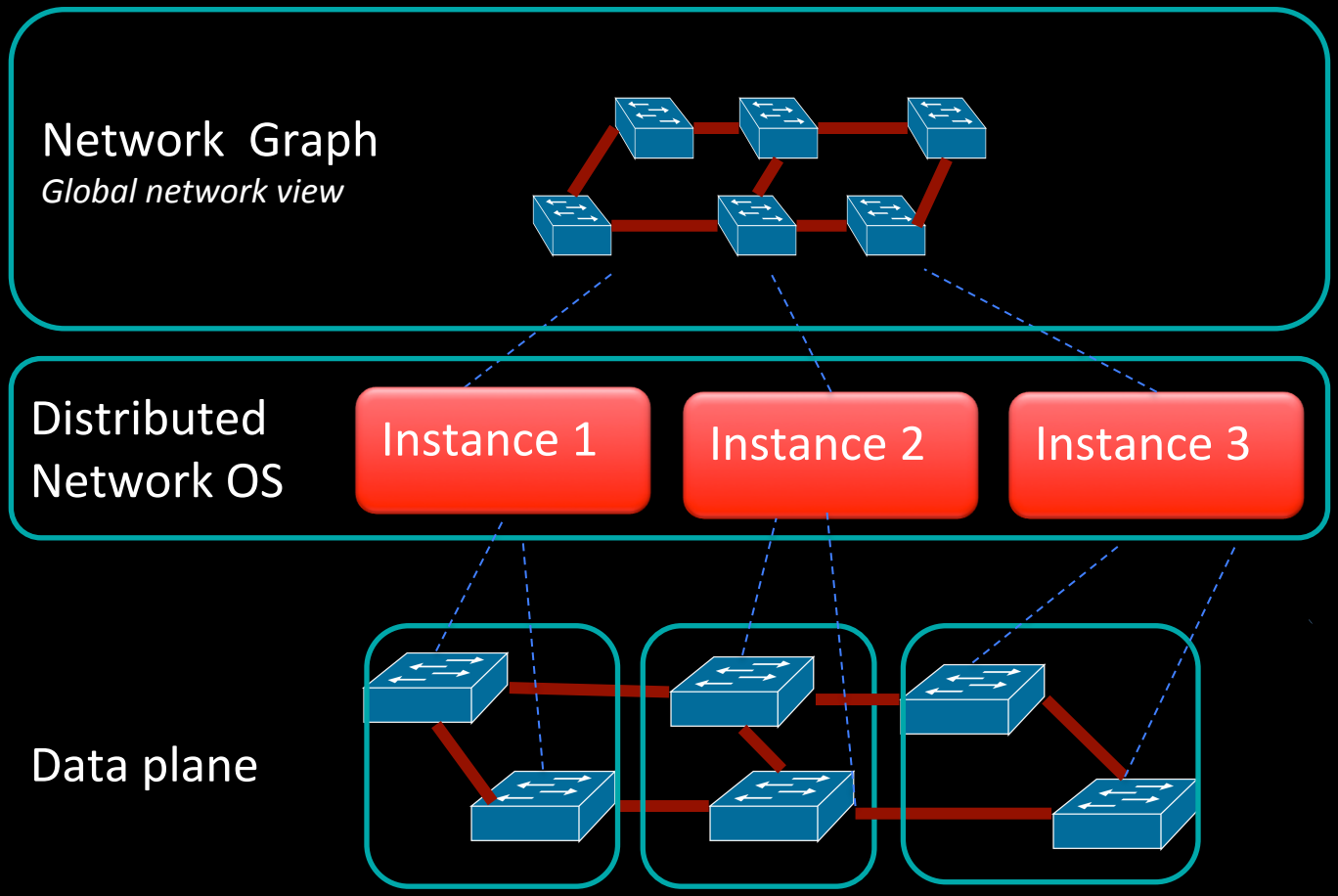
Instance 3

OF Controller

Floodlight



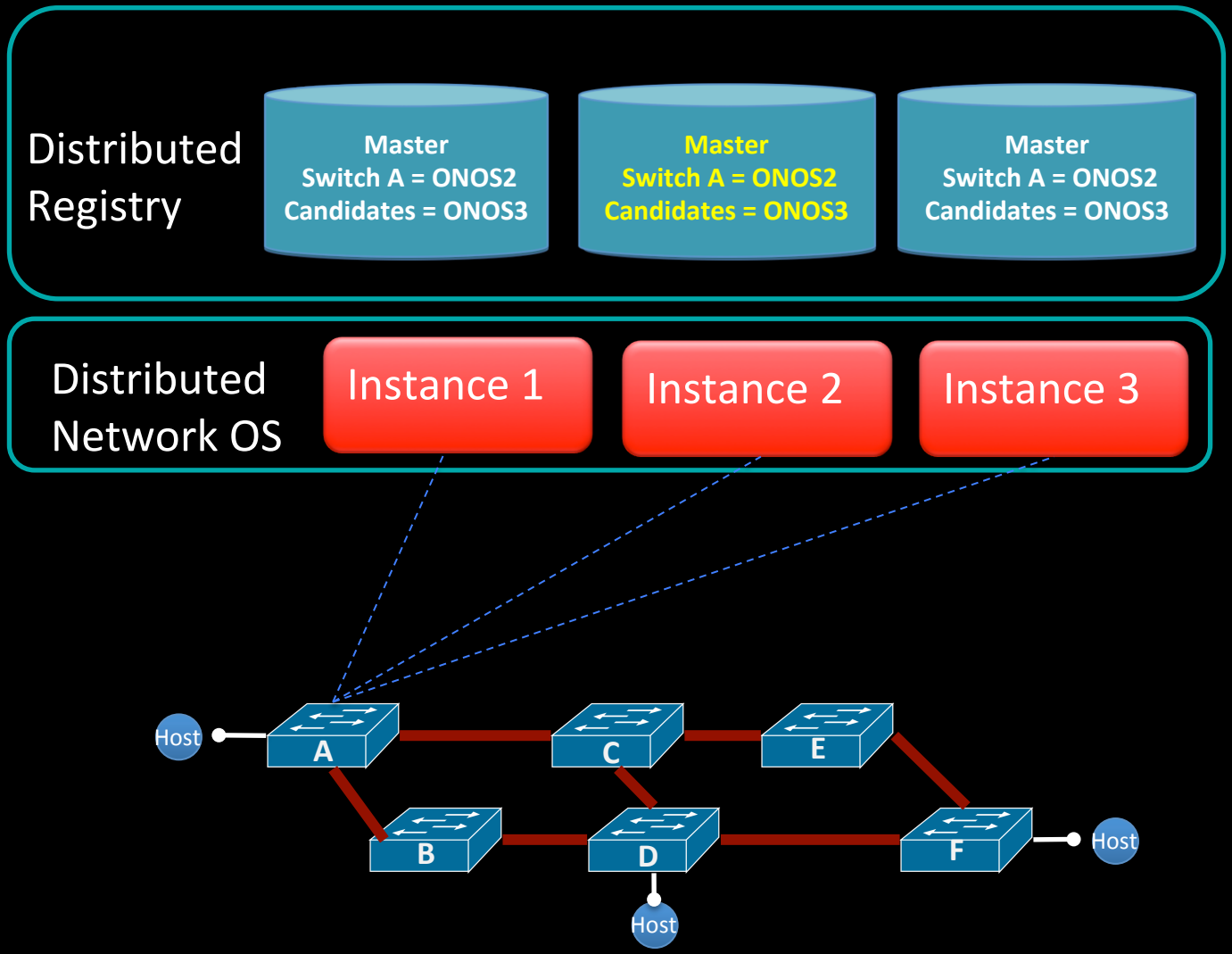
ONOS Scale-Out



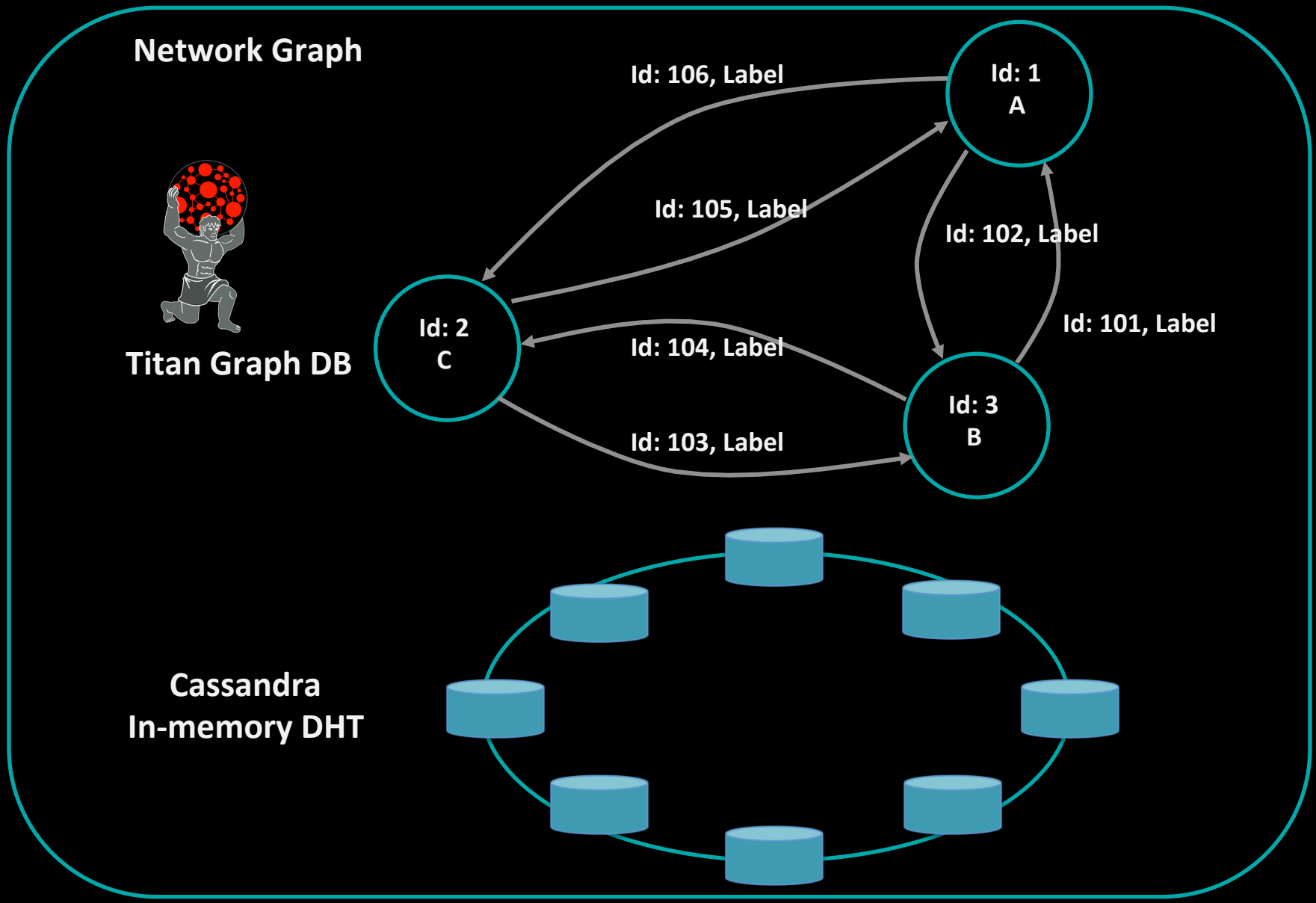
An instance is responsible for maintaining a part of network graph

Control capacity can grow with network size or application need

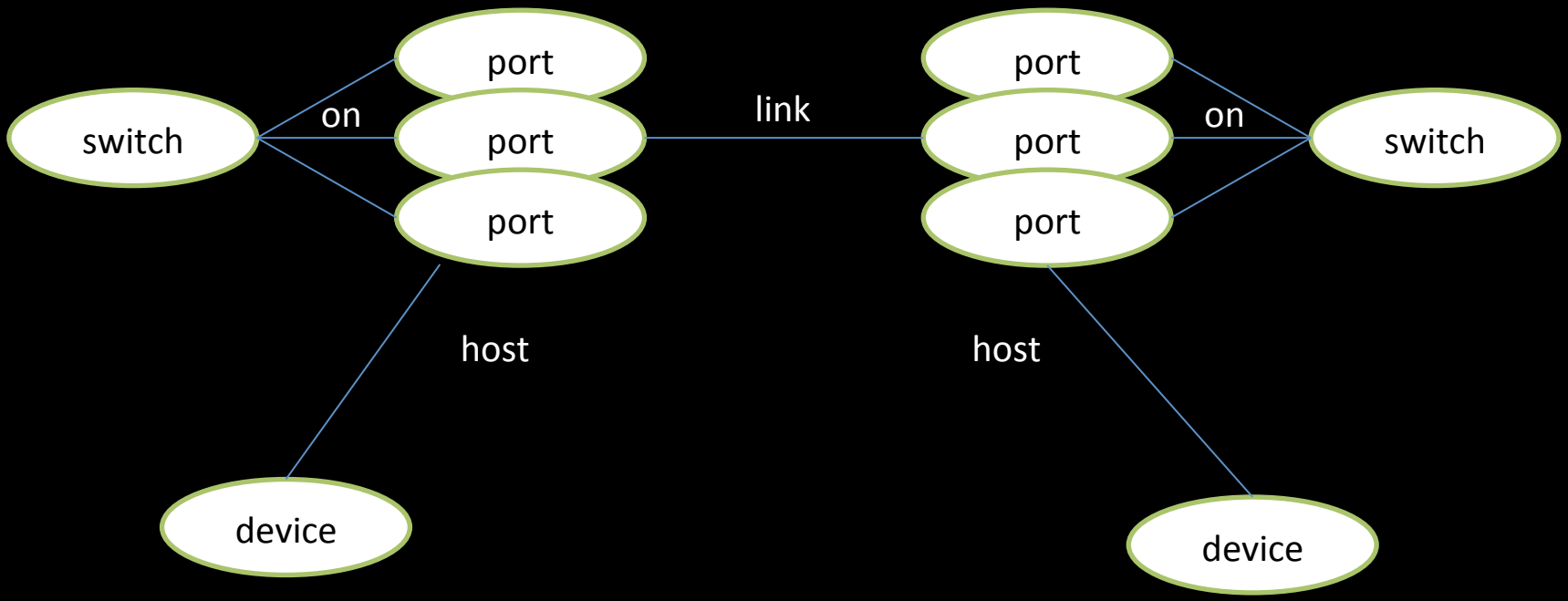
ONOS Control Plane Failover



ONOS Network Graph Abstraction

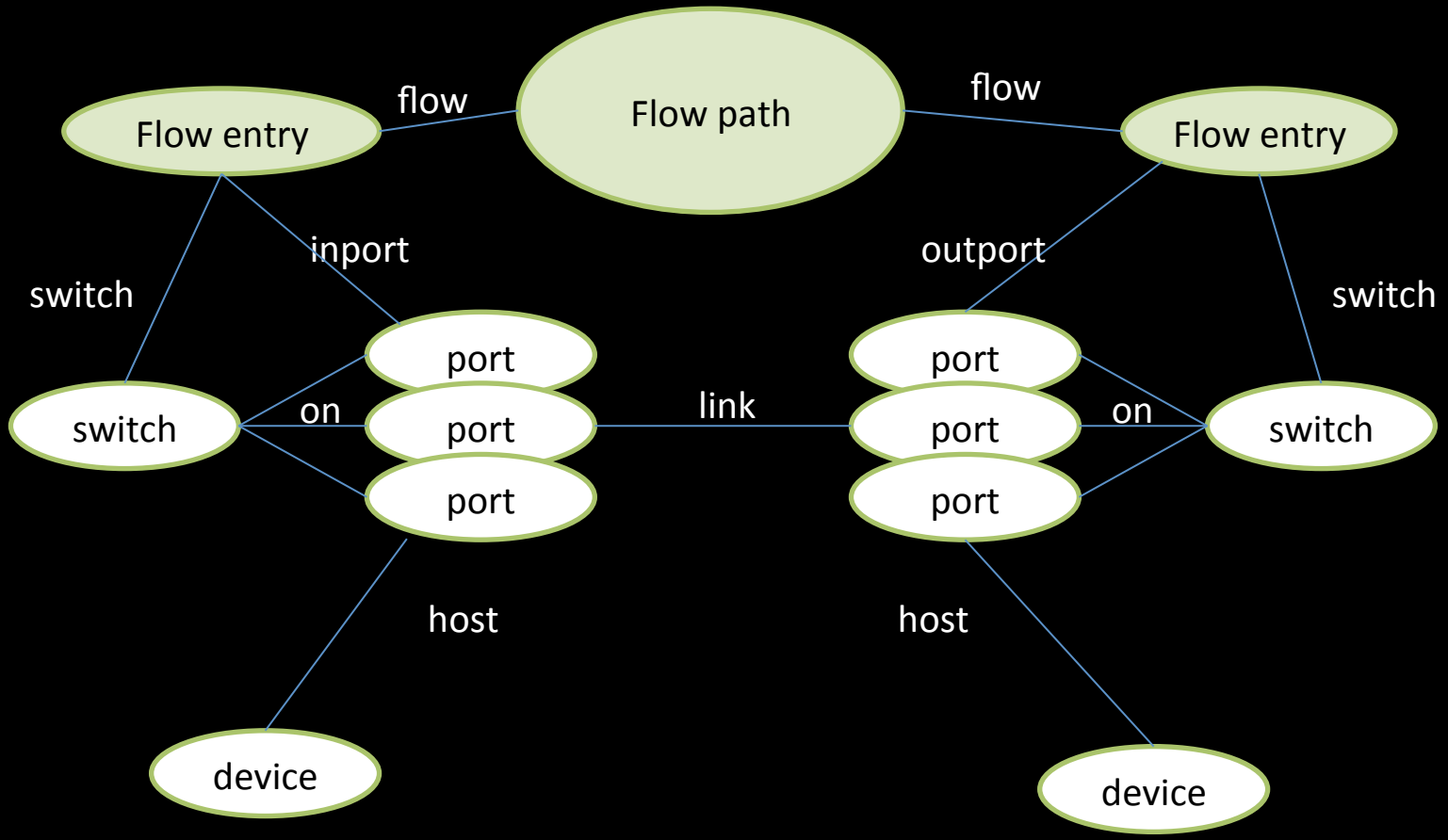


Network Graph



- **Network state is naturally represented as a graph**
- **Graph has basic network objects like switch, port, device and links**
- **Application writes to this graph & programs the data plane**

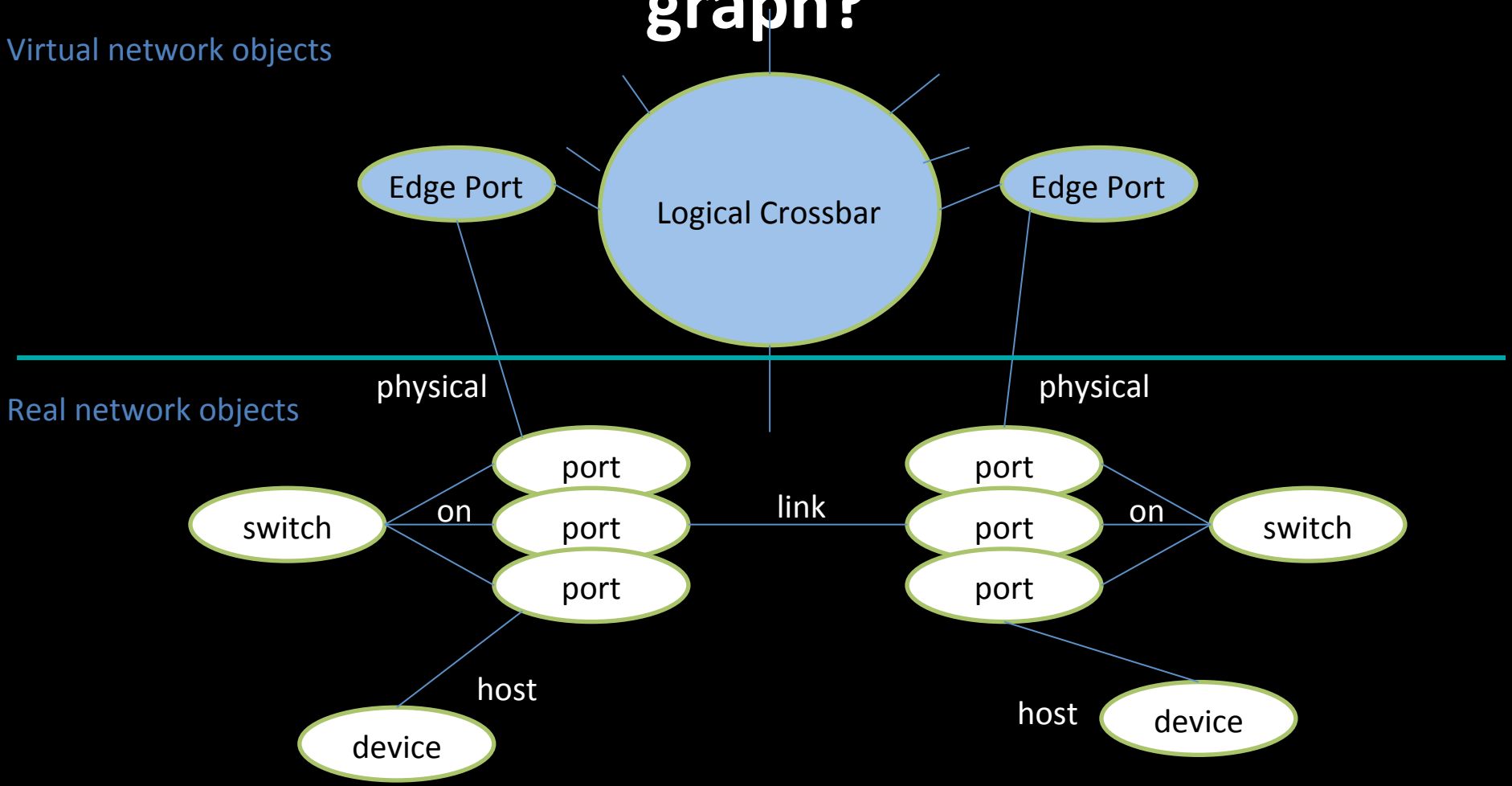
Example: Path Computation App on Network Graph



- **Application computes path by traversing the links from source to destination**
- **Application writes each flow entry for the path**

  Thus path computation app does not need to worry about topology maintenance

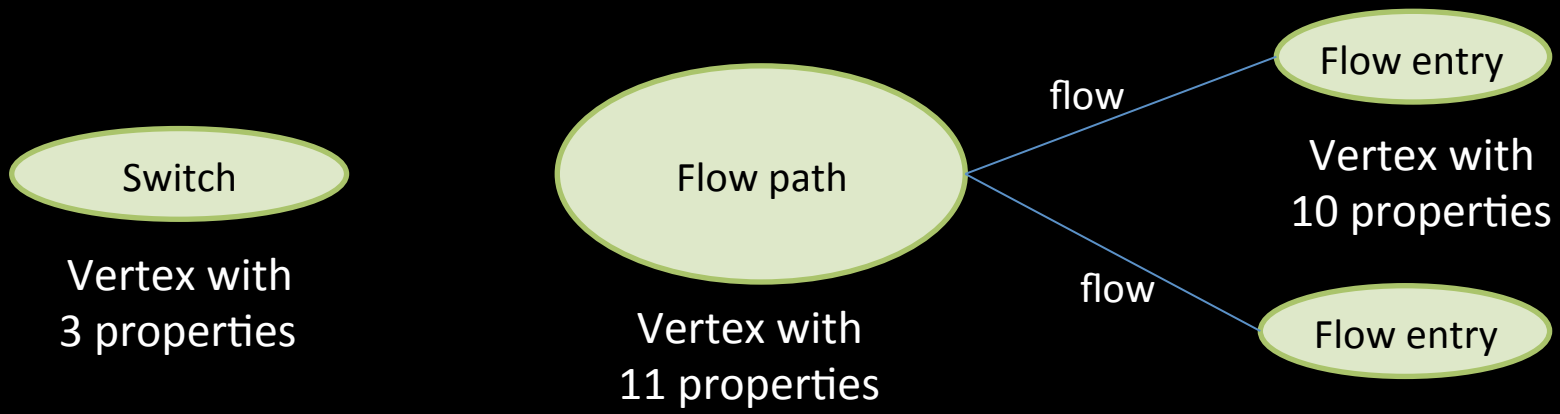
Example: A simpler abstraction on network graph?



- **App or service on top of ONOS**
- **Maintains mapping from simpler to complex**

Thus makes applications even simpler and enables new abstractions

Network Graph in Titan and Cassandra



Vertex represented as Cassandra row

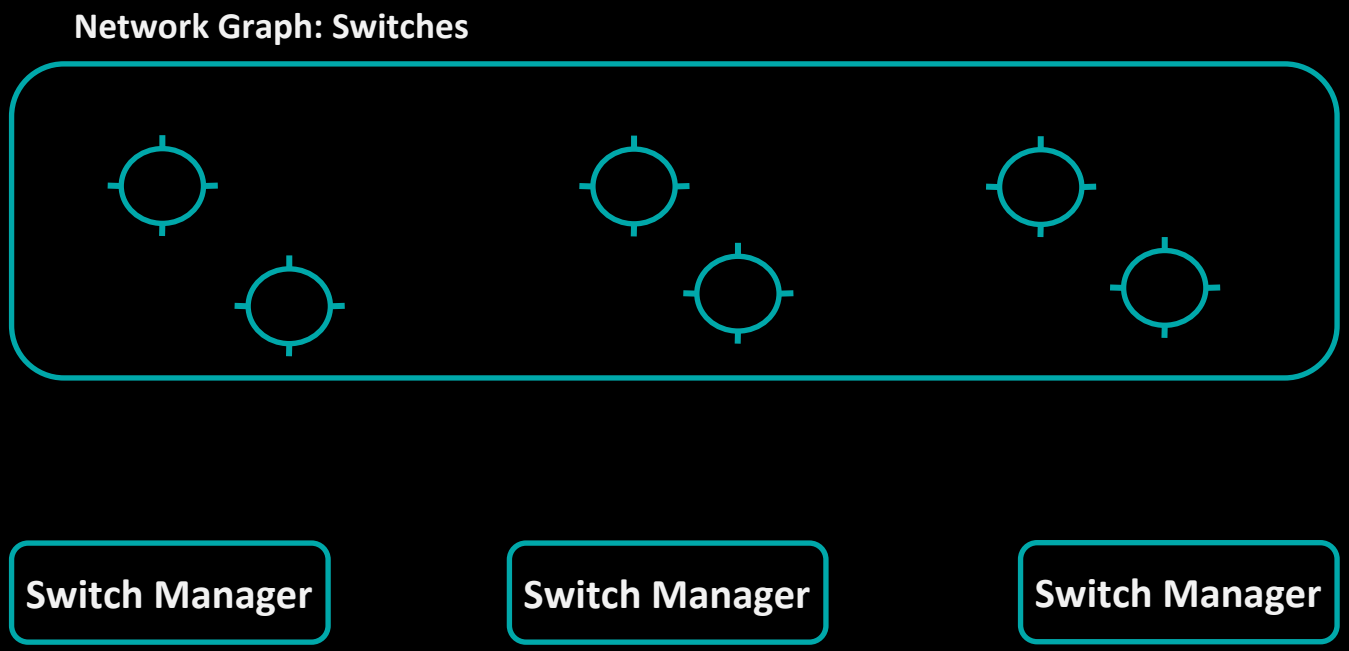
Property (e.g. dpid)	Property (e.g. state)	Property	…	Edge	Edge

Row indices for fast vertex centric queries

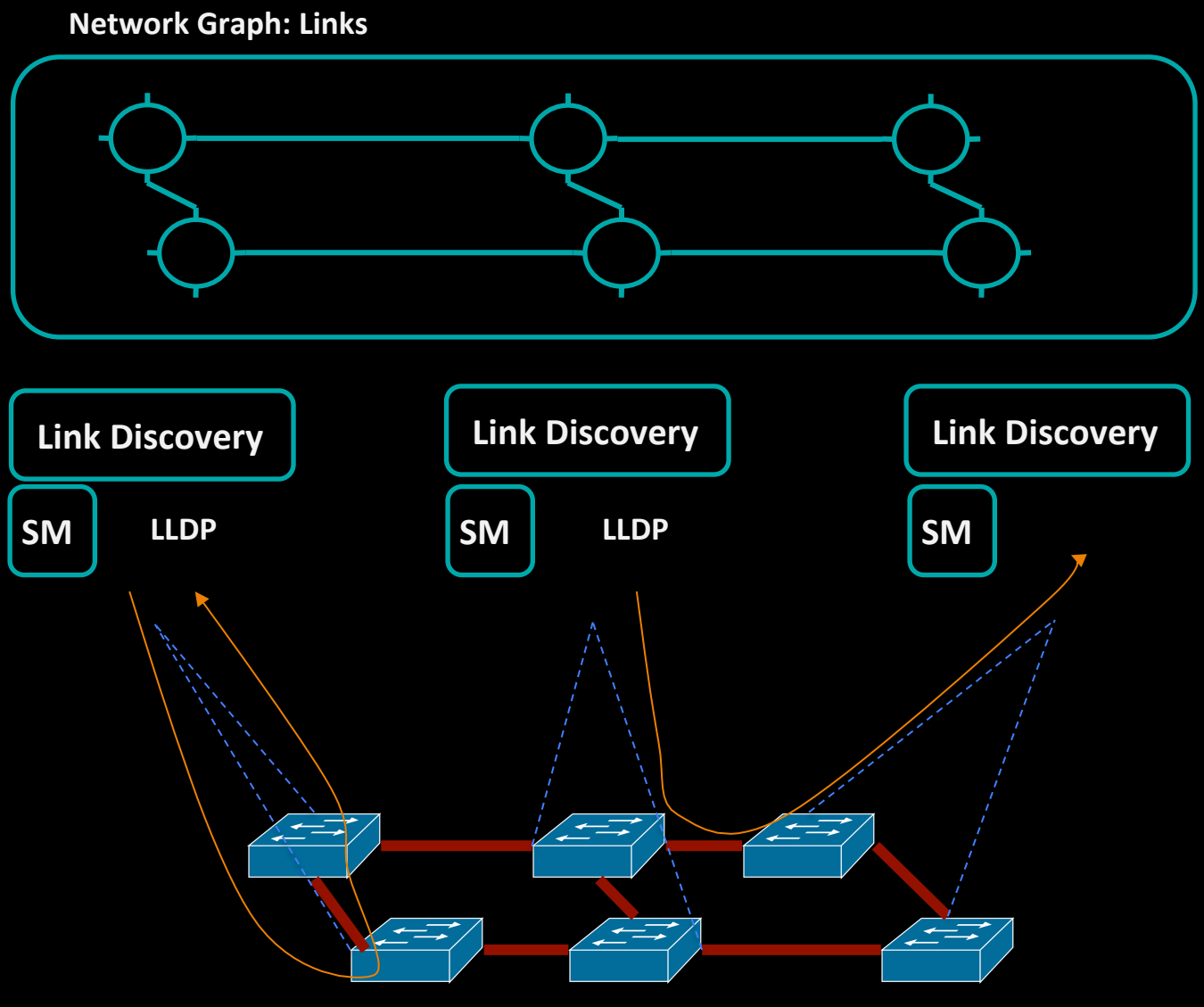
Edge represented as Cassandra column

Column				**Value**	
Label id + direction	Primary key	Edge id	Vertex id	Signature properties	Other properties

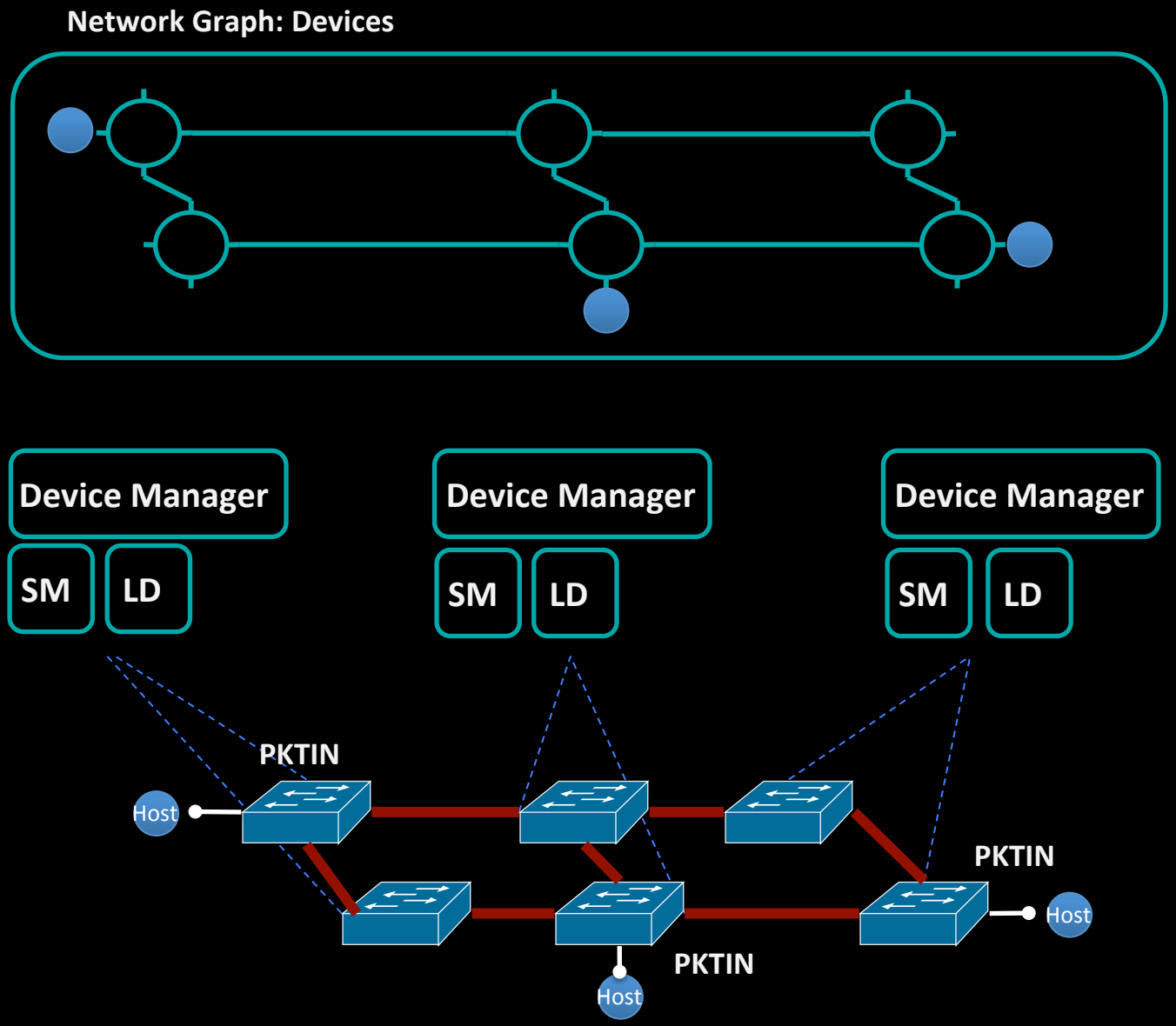
Network Graph and Switches



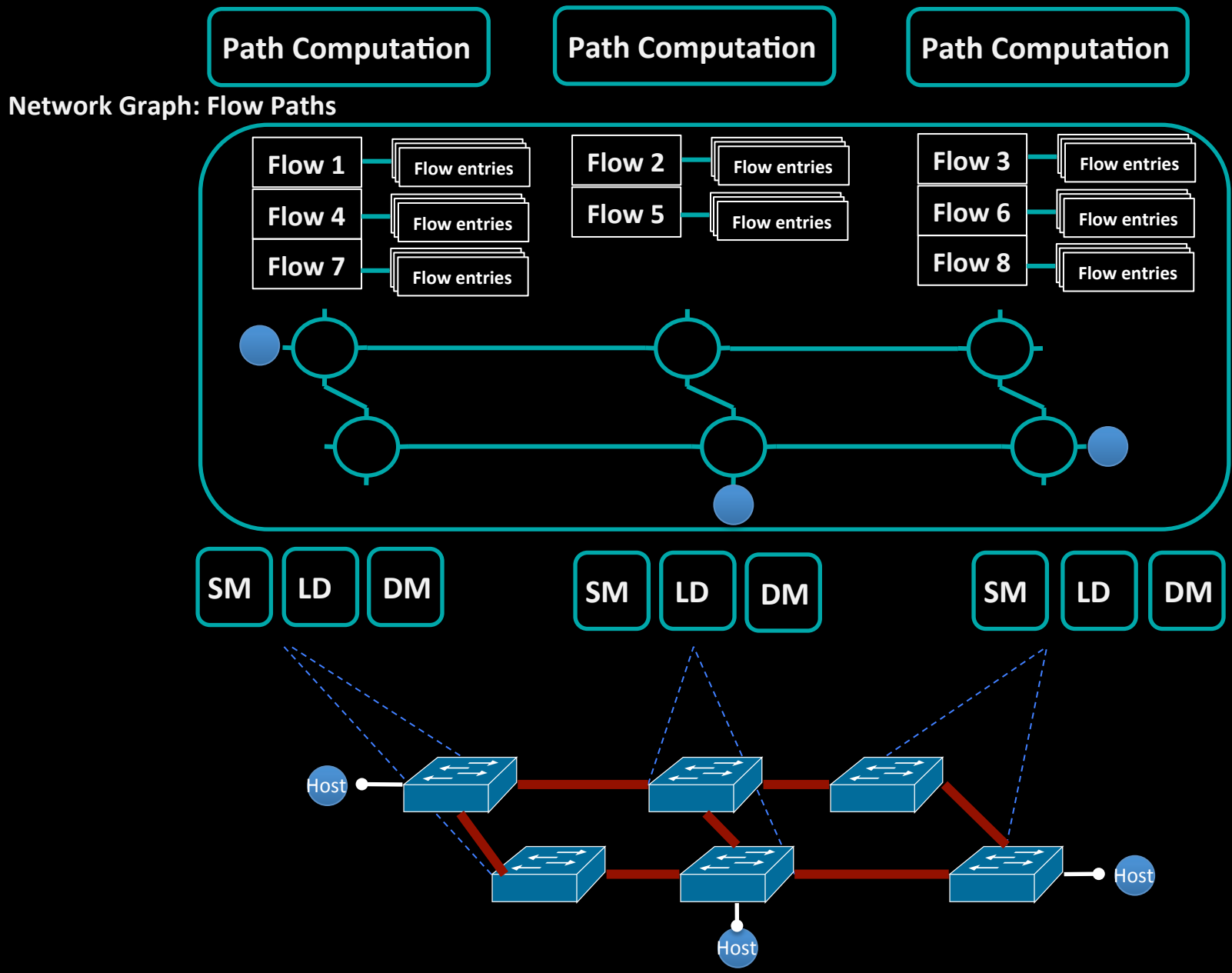
Network Graph and Link Discovery



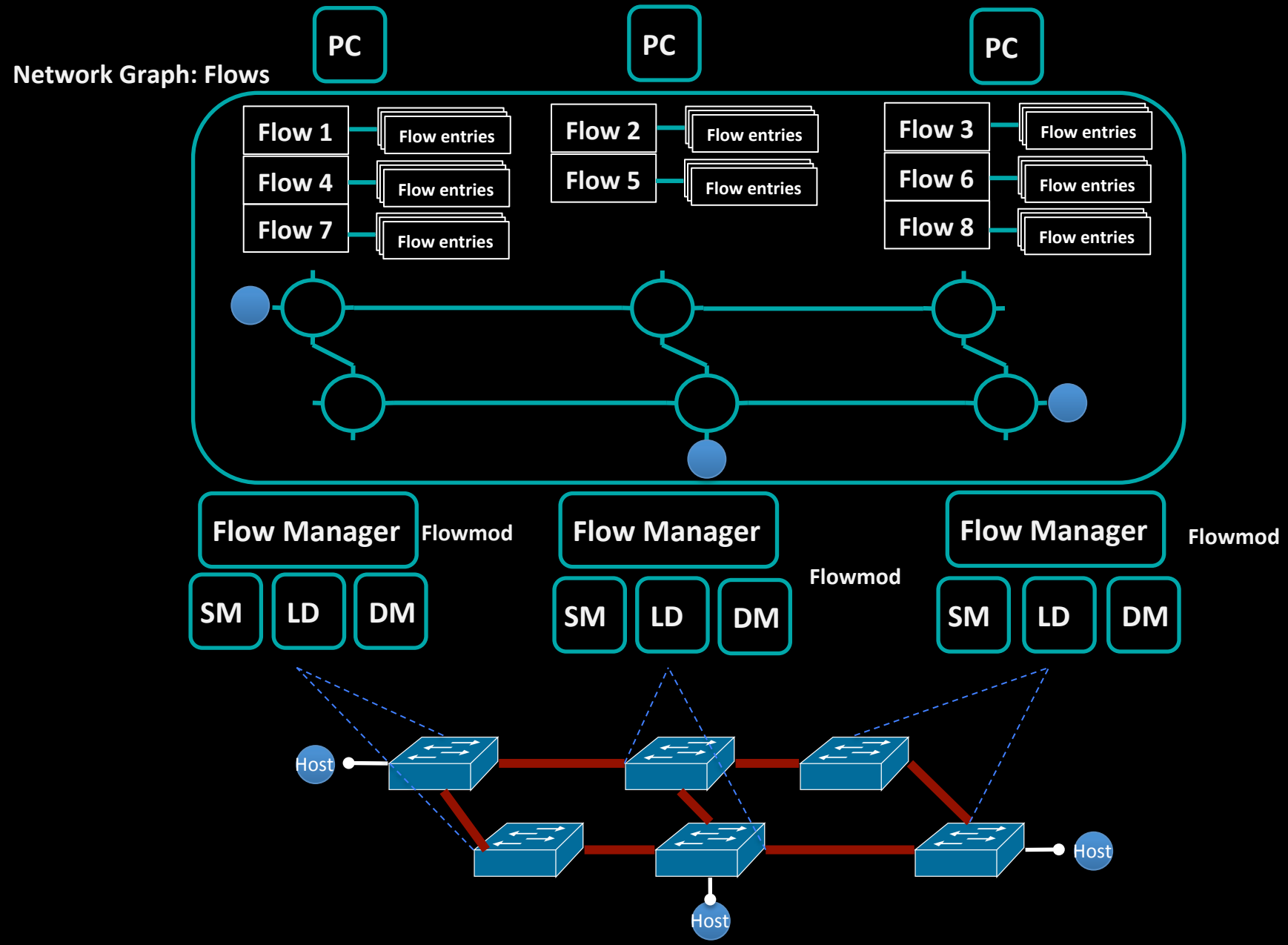
Devices and Network Graph



Path Computation with Network Graph



Network Graph and Flow Manager



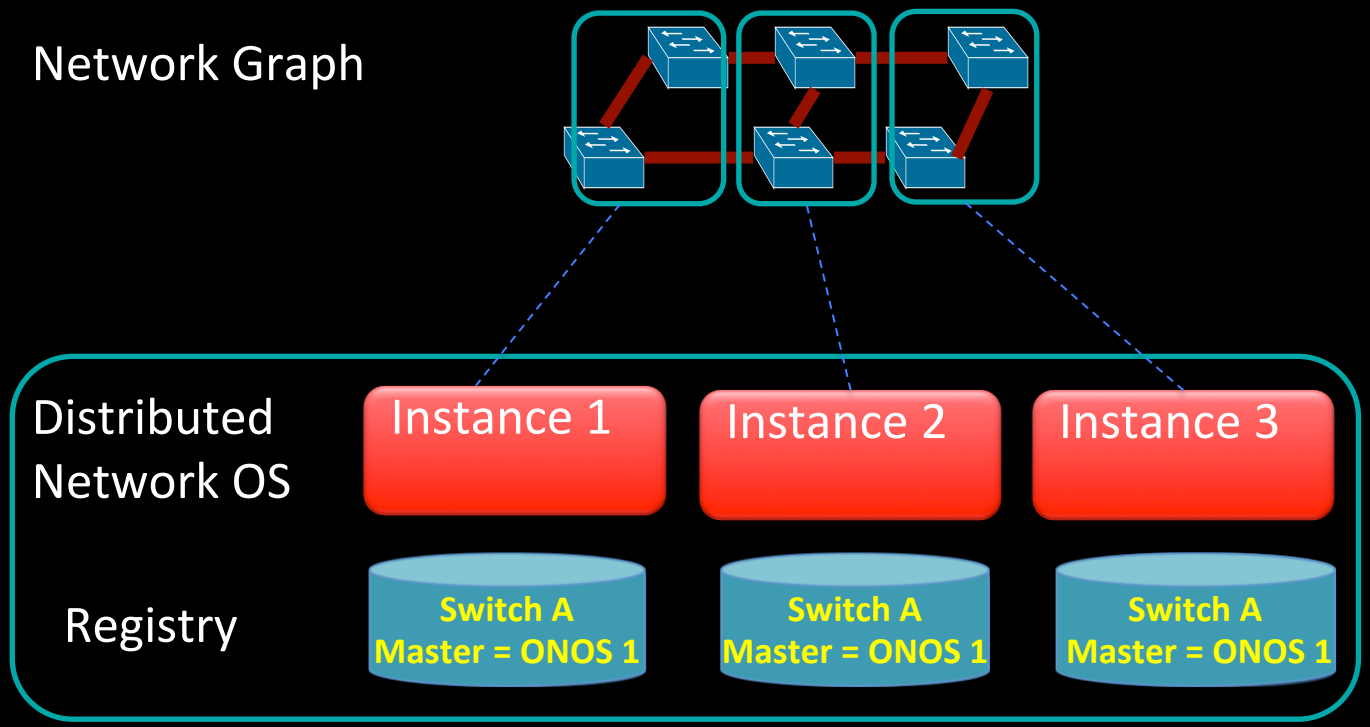
Consistency Definition

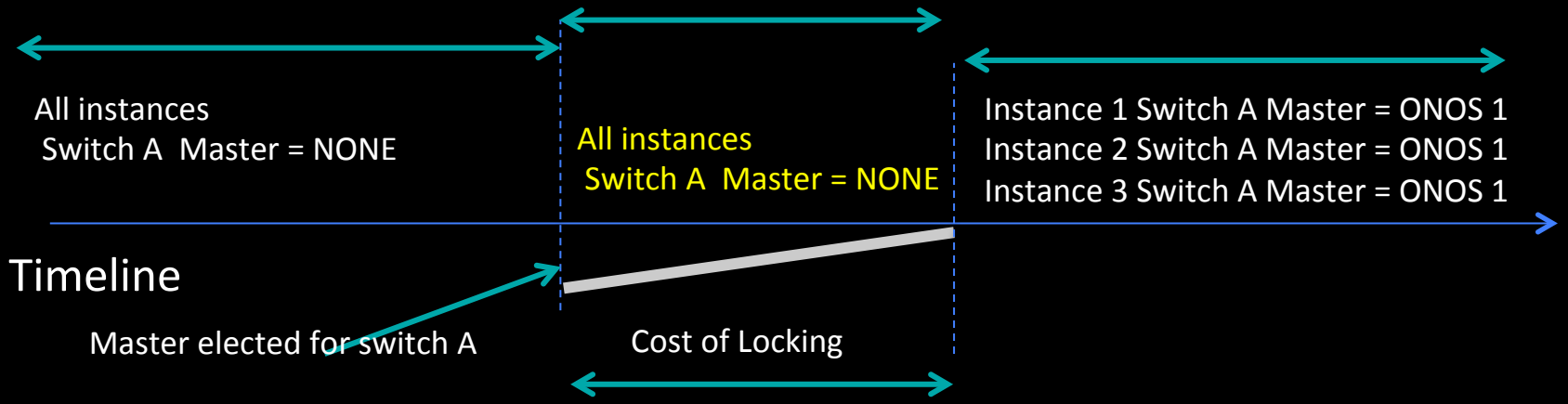
- **Strong Consistency: Upon an update to the network state by an instance, all subsequent reads by any instance returns the last updated value.**

- **Strong consistency adds complexity and latency to distributed data management.**

- **Eventual consistency is slight relaxation – allowing readers to be behind for a short period of time.**

Strong Consistency using Registry

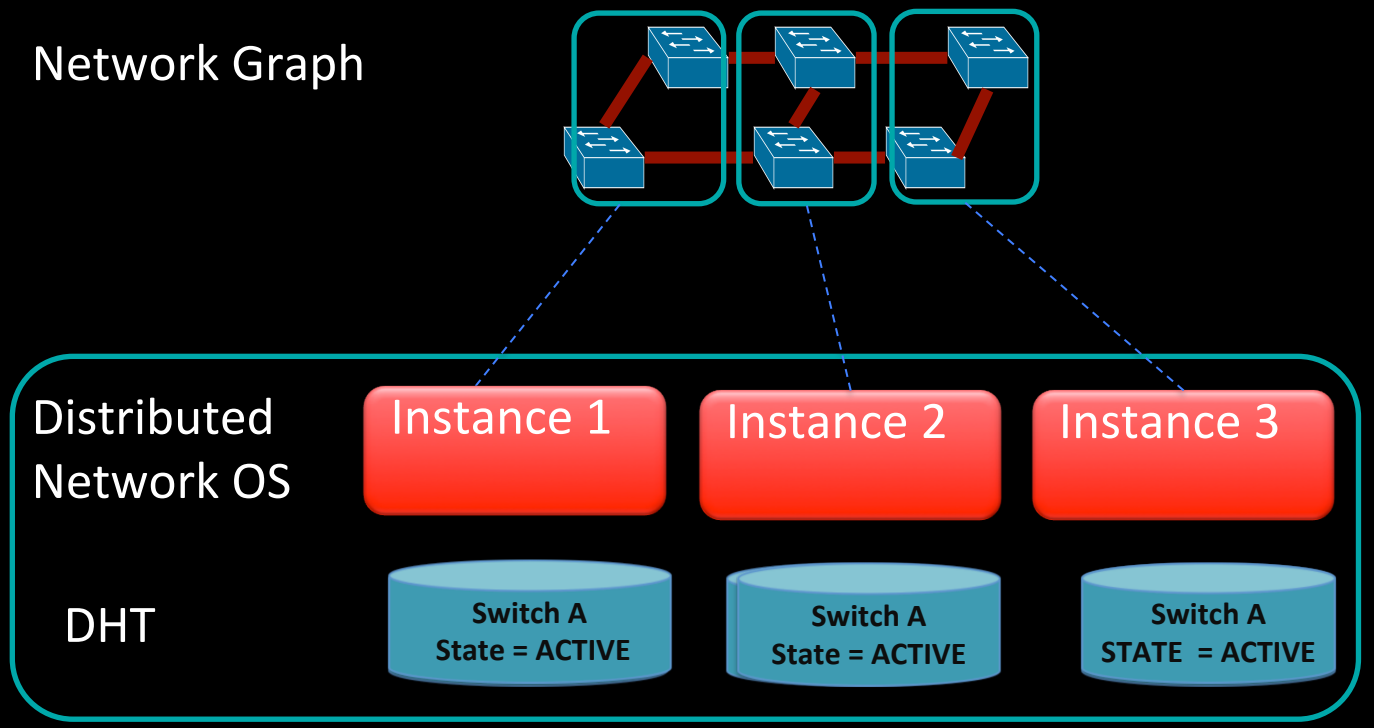


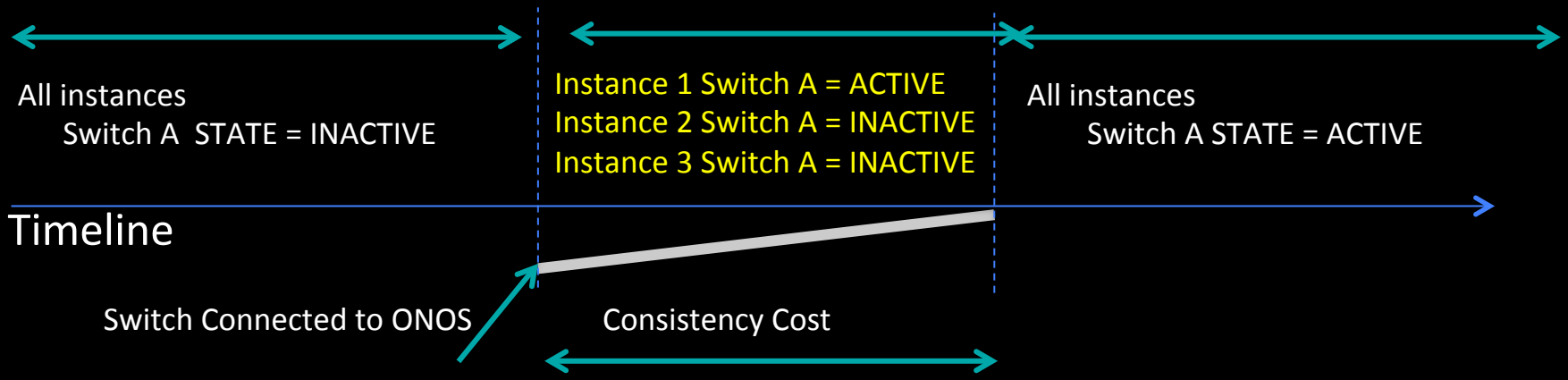


Why Strong Consistency is needed for Master Election

- Weaker consistency might mean Master election on instance 1 will not be available on other instances.
- That can lead to having multiple masters for a switch.
- Multiple Masters will break our semantic of control isolation.
- Strong locking semantic is needed for Master Election

Eventual Consistency in Network Graph





Cost of Eventual Consistency

- Short delay will mean the switch A state is not ACTIVE on some ONOS instances in previous example.
- Applications on one instance will compute flow through the switch A while other instances will not use the switch A for path computation.
- Eventual consistency becomes more visible during control plane network congestion.

Why is Eventual Consistency good enough for Network State?

- Physical network state changes asynchronously
  - Strong consistency across data and control plane is too hard
  - Control apps know how to deal with eventual consistency

- In the current distributed control plane, each router makes its own decision based on old info from other parts of the network and it works fine

- Strong Consistency is more likely to lead to inaccuracy of network state as network congestions are real.

What is Next for ONOS

ONOS Core

- Performance tuning
- Events, callbacks and publish/subscribe API
- Expand graph abstraction for more types of network state

ONOS Apps

- Intelligent control plane: SDN-IP
- Demonstrate two new use cases by end of the year

Community

- Limited availability release in Q3
- Build and assist developer community outside ON.LAB
- Support deployments in R&E networks

ON.LAB

www.onlab.us