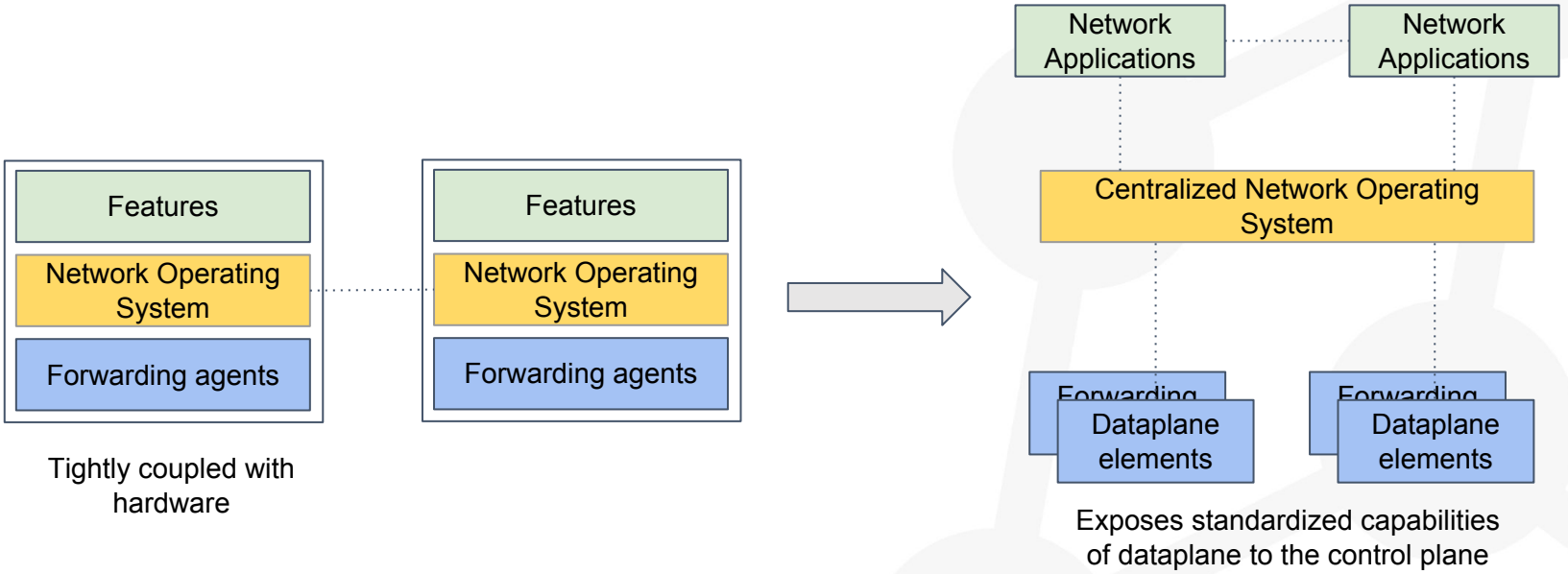
OPEN NETWORKING
SUMMIT 2016
MARCH 14-17, 2016 | SANTA CLARA, CA

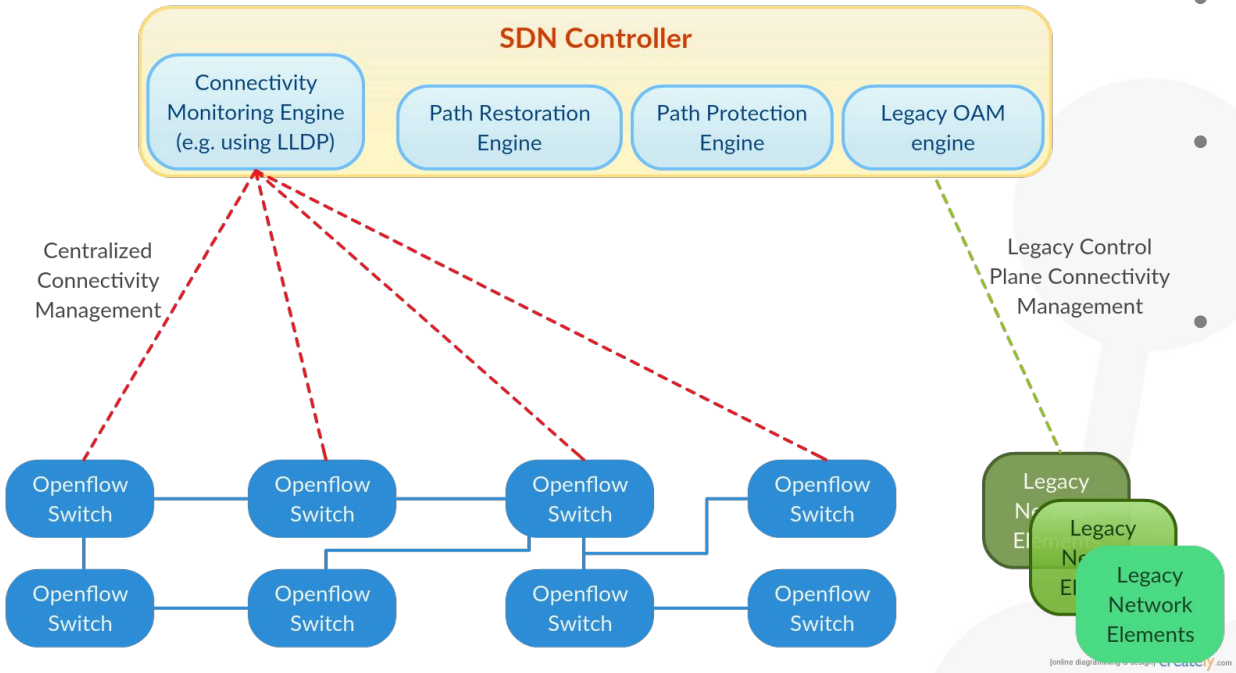
Path protection and Failover strategies in SDN networks

Rashmi Pujar, Icaro Camelo
Inocybe Technologies

SDN drastically changes the way networks are managed



SDN networking paradigm



- High reliability is a key requirement in carrier grade networks.
- State-of-art SDN network can have both Openflow and Legacy network elements being managed by SDN controller.
- Ability to perform Operations, Administration, and Maintenance (OAM) in the switch is a must in such networks since reliability and failover with minimum traffic disruption is a crucial requirement.

The Problem Statement

- Path Protection and network recovery from failure is critical aspect of Network Management.
- Directly impacts network quality and ability of service providers to meet their SLAs.
- With SDN evolving towards adoption in production covering the basic aspect of protection is the key.
- How ready is SDN for this?

Points of Failures in SDN

In SDN network, failures of controller-to-switch connection can be mainly attributed to two main network failures:

- Link or Node failures
- With decoupling of data plane and control plane SDN controller itself is an additional source of failure.

The scope of this talk is mainly about handling Link and Node failures.

Retrospective: Handling failures in traditional IP networks

In Traditional distributed IP networks possess inherent resiliency:

- Each node has network topology data and can autonomously take forwarding decisions.
- Execution of routing protocols (BGP, OSPF, IS-IS) triggers network convergence, dedicated link monitoring protocols (Link state PDUs, BFD, LLDP)
- A distributed network could be recovered in 50ms or less.
- Path Protection schemes: to allow a transient solution to reduce packet loss while network convergence process completes.

In a nutshell Network recovery is a well-understood topic

Challenges: Dealing with failures in Openflow SDN approach

In SDN networks, Openflow switches lack a local control plane.

- At most they can identify a failed link but have to wait on the controller to establish alternate routes to react to any topology changes.
- Much younger paradigm

Challenges: Dealing with failures in Openflow SDN approach

Recovery mainly comprises of:

- Mechanism for controller to detect failures.
- Controller reacting to it by programming new network state to affected switches. Optionally, relax the separation of control plane operations by including connectivity monitoring OAM in the switch.
- Network recovery time is affected by communication delay between controller-to-switch and delay associated with computation of new network state.

How Opendaylight tackles this?

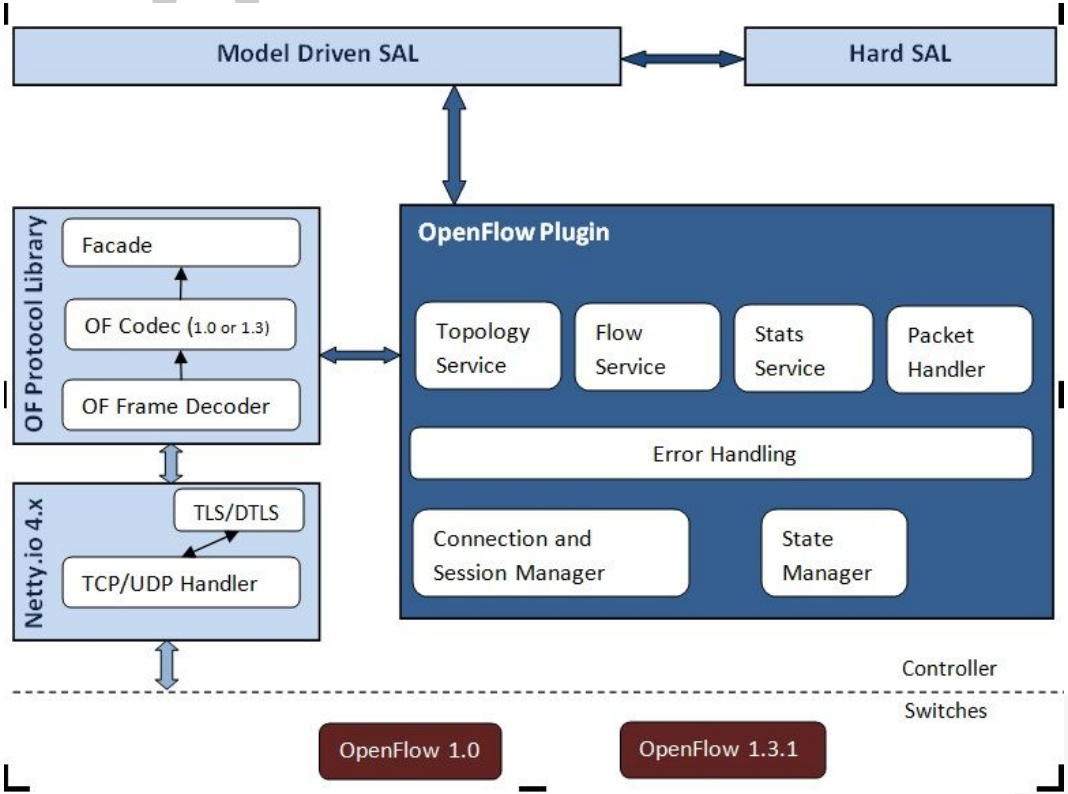
- Link and Node monitoring, Building a network topology view is handled by the Openflow Plugin Project applications called Network Service Functions (NSF).

- Path calculation engines such as BGP-PCEP project, topoprocessing project

Software Failover in Opendaylight

- Controller is solely responsible to detect network failure.
- Could use OAM tools like LLDP (Link Layer Discovery Protocol).
- Path protection: compute disjoint paths using path calculation engines (for example: Suurballe Algorithm).
- Delays associated with S/W Failovers are high.
- A centralized monitoring model poses scalability limitations as it create a lot of load in control plane thereby might overwhelm the controller with monitoring messages.

Openflowplugin NSF applications



- **Topology Manager** – Builds network topology using LLDP speaker
- **Inventory Manager** – Handles Southbound devices
- **Statistics Manager** – Collects statistics information like flows, ports, table stats
- **Forwarding Rules Manager** – Installs flows on Southbound devices

Path calculation engine

- Opendaylight's topoprocessing project includes Suurballe algorithm implementation to calculate disjoint paths.

- The main idea of Suurballe algorithm is to use Dijkstra's algorithm to find one path, to modify the weights of the graph edges, and then to run Dijkstra's algorithm a second time.

Hardware Failover

Incorporate fast-failover paths into switch forwarding tables.

- Done in hardware with OAM support (802.1ag link monitoring, BFD) and Openflow Group Tables: FAST FAILOVER - could be achieved using Openflowplugin to write flows to group tables and use OVSDB project to provision CFM/BFD.
- Pre-computed flows based on network information
- Relieves load from controller
- Lower delays, can achieve fast network restoration
- Implementation strategies for future

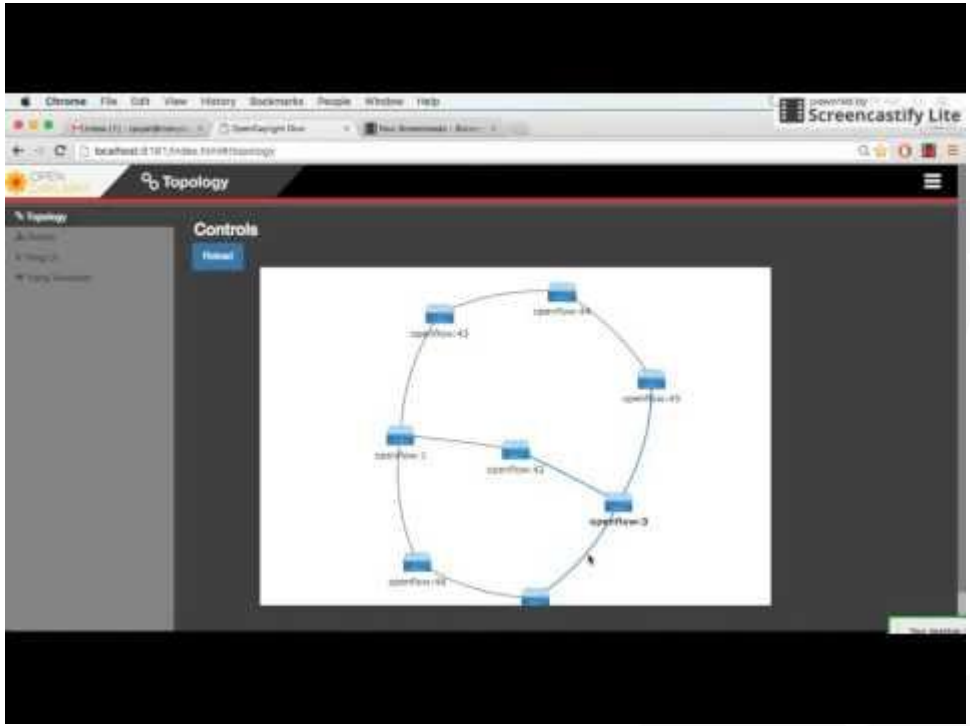
Hardware v/s Software Failover

- Delays associated with S/W failover.
- Using OAM support on vSwitch can cause control plane to get chatty.
- Fast Failover tables are pre-computed and rules are based on limited local networks, hence react only to local failures.
- End-to-end connection protection might not be realized with Openflow today. BFD support in Openflow is limited.

Demonstration of Suurballe implementation in Opendaylight

This was done in context of a use-case to implement MPLS VPNs using Intent Framework in Opendaylight. Additionally, slow-reroute path protection was provided to ensure end-to-end connectivity on Link or Port failures.

Future, work will focus on adding fast-reroute protection.



https://www.youtube.com/watch?v=wSmxJ7binSQ

Questions?