ICONA
Inter Cluster ONOS Network Application

CREATE-NET, CNIT/University of Rome Tor Vergata, Consortium GARR
The evolution of the Network Control Plane

- Standard networks:
  - Control and Forwarding Planes are embedded in the physical devices
  - Distributed system:
    - very reliable
    - requires standard protocols to share information between the devices
    - high configuration complexity
The evolution of the Network Control Plane

- **Software-defined networks:**
  - The Control Plane is logically centralized and located outside the devices
  - OpenFlow, NetConf, OVSDB, … can be used as communication protocols
The evolution of the Network Control Plane

• But the SDN Control Plane has still to be:
  - SCALABLE
  - RELIABLE
  - FAULT-TOLERANT

• So in a real scenario we probably want something like:
Open Network Operating System (ONOS)

- Discover physical topology
- React to network events
- Program the devices leveraging on different protocols
Open Network Operating System (ONOS)

- Maintain the distributed datastores
- Elect the master controller for each network portion
- Share information with the adjacent layers.
Open Network Operating System (ONOS)

- Intent Framework
  - Abstraction of the network
  - Interface for applications to interact and program the NOS
Open Network Operating System (ONOS)

- Java OSGi container in which third-party applications can be deployed
Open Network Operating System (ONOS)
ONOS
Pros
• Open source community effort with a number of partners including ON.Lab, AT&T, NEC and Ericsson among the others
• Distributed architecture
• Multiple controller instances share distributed data stores
• Eventual consistency
• Achieves scalability and resiliency
Cons
• All control plane instances must be close enough to have negligible communication delays
• Single datacenter
• LAN network
Coming to a Service Provider’s scenario

- GÉANT OpenFlow network
- 5 European Cities
- Single administrative domain
- Shared infrastructure for Experimenters
Control Plane Design

• Where can we place the ONOS control plane?
• We need to consider the latency in case of data plane failures…
• Amsterdam
  • Max one-way latency is 38 ms
Control Plane Design

• Where can we place the ONOS control plane?

• We need to consider the latency in case of data plane failures…

• Zurich
  • Max one-way latency is 27 ms

• And so on…
Control Plane Design

• What about a GEOGRAPHICALLY distributed control plane?

• In this scenario the delay between data and control planes is negligible!
ICONA: Inter Cluster ONOS Network Application

- ICONA partitions the Service Provider's network into several geographical regions, each one managed by a different cluster of ONOS instances.

- The network architect can select the number of clusters and their geographical dimension depending on requirements.

- The communication between different ICONA instances, both intra and inter-cluster, is based on Hazelcast, a multicast event-based Java Messaging System.
ICONA internal modules

- **Topology**: discovers the data plane elements, reacts to network events, shares relevant information about links and devices (with the relevant metrics) with other clusters.

- **Backup**: a partial backup path is computed and pre-installed in the data plane to minimize the rerouting process in case of data plane failures.

- **Service**: creation/deletion/modification of L2 pseudo-wire tunnels and MPLS VPN overlay networks.
Hazelcast channels
- **intra channel** devoted to the intra-cluster communication (e.g. local ONOS cluster)
- **inter channel** to share information to other clusters:
  - up-to-date view of the network
  - devices configuration

**InterLink**
- Physical connection between two clusters.
- Each IL is shared with all the clusters tagged by some metrics, such as the link delay, available bandwidth and number of flows crossing the link.
And now, a demo over the GÉANT OpenFlow testbed

- 3 ICONA clusters
- 6 ONOS instances
- 15 OpenFlow switches
- 5 VMs (1 server and 4 clients)
- 26 links