Inter-controller Traffic in ONOS Clusters for SDN Networks

Abubakar Siddique Muqaddas, Andrea Bianco, Paolo Giaccone

Guido Maier

13th Italian Networking Workshop: San Candido, Italy
Software Defined Networking

Well known advantages

• high flexibility (vendor agnostic)
• programmability
• centralized view of the network state
• simplify development of network applications

One of the many concerns

• scalability for large networks
Distributed controllers

Motivation

• fault-tolerance and resilience
• load balancing, thus higher scalability for large networks

Critical issue

• how the network state is distributed across the controllers to allow a centralized logical view
Control-plane in distributed controllers

- must be *in-band* in large networks (e.g. WAN)
- switch-controller traffic
  - standard protocols (e.g. OpenFlow)
Inter-controller traffic

• needed to coordinate distributed controllers
• ad-hoc protocols on the east-west interfaces
• support for consistency protocols
  – shared data structures
  – different models for consistency
Consistency models

• assume a shared table=\((key, value)\)

• *strong consistency*
  – any read(key) returns always the same value

• *eventual consistency*
  – any read(key) returns eventually the same value
    • after some transient time, the same value

• adopted model affects heavily
  – the mechanisms to distribute and update the data
  – the reactivity of the SDN controllers perceived by the network devices
  – the correct behavior of the network
Our contribution

• devise an experimental model to evaluate the required bandwidth for inter-controller traffic
  – useful to design and dimension the transport network for the control plane
  – we neglect switch-controller traffic
    • complementary to inter-controller traffic
    • can be independently evaluated based on the network application, flow dynamics, and adopted protocol (e.g. OpenFlow)
Distributed SDN controllers

**ONOS**
- focused mainly on service providers and WANs
- fault tolerance and state distribution across controllers
- developed by On.Lab and supported by network operators (e.g., AT&T) and by vendors (Ciena, NEC, Huawei)

**OpenDaylight**
- primarily focused on data centers but suitable also for WANs
- one controller designed to rule all the others
- internal abstractions structured to be compatible to any functionality
- supported by the Linux Foundation and by many IT industries (ADVA, Cisco, Ciena, Corian, etc.)
Consistency in controllers

ONOS
- eventual consistency
- network topology
- flow rules
- flow statistics
- strong consistency
- switch-controller mapping
- distributed locks

OpenDaylight
- strong consistency
- all shared data structures
ONOS consistency algorithms

**Anti-entropy**

- eventual consistency
- updates are local in primary controller and propagates in the background with gossip algorithms
  - every 5 sec, each controller picks at random another controller, compare replicas and reconcile differences based on timestamps

**RAFT**

- strong consistency
- updates are centralized to the leader
- an update is committed whenever a majority of controllers acks to the leader
Methodology

• mininet to emulate any test network topology
  – isolated
  – linear
  – star
• dedicated LXC container for each ONOS instance
  – ONOS 1.2 Cardinal (rel. May 2015)
• traffic sniffer to evaluate the inter-controller traffic
Model for inter-controller traffic

Input
- switch network topology and domain
  - number of switches
  - number of intra-domain links
  - number of inter-domain links

Output
- analytical formulas for the inter-controller traffic
ONOS inter-controller traffic

• Memory of past network states
  – add and removing switches changes the baseline bw
  – due to `tombstone'' data kept for faster recovery in case of failures

• Each experiment required to reboot the containers to avoid tombstone traffic
Scenario with 2 controllers

- linear topology connected to controller A
Scenario with 3 controllers

- linear topology connected to controller A
Analytical model for 3 controllers

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{ij}, (B^g_{ij}, B^h_{ij})$</td>
<td>generic unidirectional bandwidth (in isolated or linear topology)</td>
</tr>
<tr>
<td>$b^g_{ij}$</td>
<td>generic zero bandwidth</td>
</tr>
<tr>
<td>$b^h_{ij}$</td>
<td>average unidirectional bandwidth per switch</td>
</tr>
<tr>
<td>$b^i_{ij}$</td>
<td>average unidirectional bandwidth per intra-domain link</td>
</tr>
<tr>
<td>$b^d_{ij}$</td>
<td>average unidirectional bandwidth per inter-domain link</td>
</tr>
<tr>
<td>$b^e_{ij}$</td>
<td>average unidirectional bandwidth per inter-domain link external to target controller</td>
</tr>
</tbody>
</table>

- $B^g_{ij}$ bandwidth $A \rightarrow B$ and $A \rightarrow C$ in isolated topology, $B \rightarrow A$, $C \rightarrow A$, $B \rightarrow C$ and $C \rightarrow B$ in isolated topology
- $B^h_{ij}$ bandwidth $A \rightarrow B$ and $A \rightarrow C$ in linear topology, $B \rightarrow A$, $C \rightarrow A$, $B \rightarrow C$ and $C \rightarrow B$ in linear topology
- $b^i_{ij}$ average bandwidth $A \rightarrow B$ and $A \rightarrow C$ per switch
- $b^d_{ij}$ average bandwidth $B \rightarrow A$, $C \rightarrow A$, $B \rightarrow C$ and $C \rightarrow B$ per switch
- $b^e_{ij}$ average bandwidth $A \rightarrow B$ and $A \rightarrow C$ per link
- $b^e_{ij}$ average bandwidth $B \rightarrow A$, $C \rightarrow A$, $B \rightarrow C$ and $C \rightarrow B$ per link

**Property 2:** In a scenario with 3 controllers $A$, $B$ and $C$, managing a generic network topology arbitrarily partitioned among the three controllers, the exchanged bandwidth can be evaluated as follows:

\[
B_{A\rightarrow BC} = 43 + 1.12 \times S_A + 0.93 \times L_A + 0.35 \times (S_B + S_C) + 0.53 \times (L_B + L_C) + 0.87 \times (L_{AB} + L_{AC}) + 0.7 \times L_{BC} \text{ [kbps]}
\]

\[
B_{B\rightarrow AC} = 43 + 1.12 \times S_B + 0.93 \times L_B + 0.35 \times (S_A + S_C) + 0.53 \times (L_A + L_C) + 0.87 \times (L_{AB} + L_{BC}) + 0.7 \times L_{AC} \text{ [kbps]}
\]

\[
B_{C\rightarrow AB} = 43 + 1.12 \times S_C + 0.93 \times L_C + 0.35 \times (S_A + S_B) + 0.53 \times (L_A + L_B) + 0.87 \times (L_{AC} + L_{BC}) + 0.7 \times L_{AB} \text{ [kbps]}
\]
Conclusions

Contributions

- experimental evaluation of the inter-controller traffic in ONOS SDN clusters
  - around 1 kbps for each network element (switch / port)
  - even if low bandwidth, this control information is critical for the correct network behavior
- highlights scalability laws
  - global inter-controller traffic grows
    - quadratically with the number of controllers
    - linearly with the number of network elements
- control plane must be carefully designed and dimensioned