Devolve-Redeem
Hierarchical SDN controllers with adaptive offloading

Rinku Shah
Mythili Vutukuru
Purushottam Kulkarni
IIT Bombay, India

3rd August, APNet 2017
Traditional network vs Software-defined network

- Simplified network mgmt
- Ease of control-plane programming
How far can SDN Controllers scale?

Openflow controllers support around 10K to 30K flows/sec *

SDN networks have flow arrival rate of 100K to 1M flows/sec**

** Kandula and others.. The nature of data center traffic: measurements & analysis. In Proceedings of IMC 2009
Controller Scaling technique - HORIZONTAL

Subset of switches assigned to each controller

Need for synchronization between controllers

Controller Scaling technique - VERTICAL

We call this technique, LSCO (Local state based compute offload)

LOCAL state egs
Flow stats
Switch mappings

Controller Scaling techniques: Abstract view

(a) Horizontal scaling
(b) Vertical scaling (LSCO)
Can Vertical Scaling perform better?

(a) Horizontal scaling
(b) Vertical scaling (LSCO)
(c) Can Vertical scaling perform better?
Key insight - GWLR state (Globally writable, but locally readable)

GWLR state examples-
1. Tunnel Id (LTE EPC)
2. MPLS label
3. Session state
4. Network Policy state
GSCO (GWLR state based compute offload)

Offload computations based on GWLR state

Should we offload all GWLR state? Synchronization cost may be high
GSCO (GWLR state based compute offload)

Offload computations based on GWLR state

Should we offload all GWLR state? Synchronization cost may be high

Centralized or LSCO or GSCO?
Key Contributions

1. **GWLR state** based offload technique
   a. **GSCO** (GWLR state based computation offload)

2. Application code is **agnostic** to scalability design

3. Framework that aids **Adaptive Offload**
   a. Designed Cost metric
   b. Implemented OVS feature for Compute Placement
Use-case: SDN based LTE-EPC application

LTE-EPC procedures considered:
1. Attach Request
2. Service Request
Devolve-Redeem Design

A
- User Input

B
- Offload Cost Metric calculator
- Enforce Offload
- Openflow Rules

C

Centralized Controller
- State management API

State synchronization

Local Controller

Local Controller
# A. User Input for LTE-EPC

**Example LTE-EPC Messages**

<table>
<thead>
<tr>
<th>Message Type</th>
<th>$N^L_R$</th>
<th>$N^L_W$</th>
<th>$N^X_R$</th>
<th>$N^X_W$</th>
<th>$N^G_R$</th>
<th>$N^G_W$</th>
<th>$N^R_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth Step 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Send UE TEID</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>UE Context Release</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Context Setup Response</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

$N^L$ : # of **Local** states accessed

$N^X$ : # of **GWLR** states accessed

$N^G$ : # of **Global** states accessed

$N^R_L$ : # of Openflow **Rules**
B. Offload Cost-metric

Cost\_mode = State-access cost + Communication cost + Synchronization cost

<table>
<thead>
<tr>
<th>Offload Mode</th>
<th>Communication Cost</th>
<th>Synchronization Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>RTT to ROOT</td>
<td>0</td>
</tr>
<tr>
<td>LSCO</td>
<td>RTT to LOCAL/ROOT</td>
<td>0</td>
</tr>
<tr>
<td>GSCO</td>
<td>RTT to LOCAL/ROOT</td>
<td>Depends on current traffic mix</td>
</tr>
</tbody>
</table>
C. Enforce Offload module

This flow should be followed for each message
Experimental Setup

Intel Xeon E312xx @ 2.16Ghz
  16 cores
  8 cores
Host: Ubuntu 14.04
Components: LXC Containers
Network: LXC Bridges
Floodlight v1.2
Openvswitch v2.3.2

Root SDN Controller
Floodlight
CPU: 1core
RAM: 8GB

Local SDN Controller

RAN
CPU: 2 cores
RAM: 8GB

DGW OpenvSwitch
CPU: 2 cores
RAM: 8GB

SGW OpenvSwitch
CPU: 2 cores
RAM: 8GB

PGW OpenvSwitch
CPU: 2 cores
RAM: 8GB

SINK
CPU: 2 cores
RAM: 8GB
1. What is the best offload scheme for a given traffic mix?

2. What is the impact of the offload choice on-
   a. Request Completion Time (Latency)
   b. Root Controller Traffic
   c. Root Synchronization Cost
Evaluation - Offload A: All GWLR state

- ATTACH <= 20%
  - GSCO = 1.4X Centralized
- 20% < ATTACH <= 60%
  - LSCO = 1.27X Centralized
- ATTACH > 90%
Evaluation - Offload A: All GWLR state

OFFLOAD CHOICE: Centralized/LSCO/GSCO

DEPENDS ON CURRENT TRAFFIC MIX

ATTACH <= 20%
GSCO = 1.4X Centralized

20% < ATTACH <= 60%
LSCO = 1.27X Centralized

ATTACH > 90%
Evaluation - Offload B: Subset of GWLR state

Performance of "Offload A" for same traffic mix (1.4X)

- **ATTACH <= 20%**
  - GSCO = 2.11X Centralized

- **20% < ATTACH <= 60%**
  - LSCO = 1.23X Centralized

- **ATTACH > 90%**
Evaluation - Offload B: Subset of GWLR state

APPLICATION PERFORMANCE ALSO DEPENDS ON WHAT SUBSET OF GWLR STATE IS OFFLOADED

Performance of “Offload A” for same traffic mix (1.4X)

ATTACH <= 20%
GSCO = 2.11X Centralized
20% < ATTACH <= 60%
LSCO = 1.23X Centralized
ATTACH > 90%
Ongoing work

- Evolving the cost metric using **dynamic parameters**

  Goals:
  - Improve accuracy
  - Reduce parameter capture & monitoring overheads

- Implement the **Online Adaptive Offload** framework
Conclusion

- **Application performance** depends on:
  - Controller Scalability design chosen
  - Subset of GWLR state offloaded

- There is need for an **Online Adaptive Offload**

- **LSCO/GSCO reduces traffic to the ROOT controller**, enabling controller scale