Investigation of Anycast Implementation in Software Defined Networking

Yulei Wu
CSTNet, Computer Network Information Center (CNIC), CAS, China
wuyulei@cstnet.cn
Contents

• Introduction

• Proposed architecture

• Design and implementation

• Simulation experiment

• Conclusion and future work
Anycasting

A paradigm of communication for service discovery that selects the best one of service providers in an anycast group as a destination

Simplify the task of service discovery in the replica deployment of an Internet service
Applications

**DNS:** 1) Realizing host auto-conf, 2) Increasing the efficiency of DNS resolution

**CDN:** 1) Optimizing the sever selection, 2) Increasing the service performance

**DCN:** 1) Mgmt of data backup, 2) operation and maintenance for cost reduction
Existing Anycast Schemes

• **IP Anycast**
  • *Weak global control capability*: The routing devices cannot be aware of the global info, and thus are hard to select the optimized routing decision.
  • *Low flexibility*: The closed and black-box behavior of routing devices make the users hard to change routing strategy, which degrades the flexibility of the configuration of anycast service.

• **Application-layer Anycast**
  • *Insensitive to topo-changes*: Anycast controller cannot obtain the topology changes, and thus cannot make the corresponding routing strategies.
  • *Low comm efficiency*: Anycast client has to communicate with controller for address resolution before initiating the communication, which degrades the efficiency of anycast service.
• Philosophy of SDN: Separate decision-making for forwarding and its corresponding actions. The controller of SDN determines the forwarding strategy in a centralized manner and the switches of SDN are only responsible for data forwarding.

• OpenFlow controller is able to agilely configure the behavior of OpenFlow switches by changing the flow table.

• The OpenFlow protocol facilitates the communications between OpenFlow controller and switches. The OpenFlow controller can be aware of the status changes of OpenFlow switches.

• The data packets do not need to be handled by the controller if being matched in the flow table, which increases the efficiency of communication.
Anycasting based on SDN/OpenFlow

- **Anycast servers** providing anycast service, **anycast client**, OpenFlow-enabled Anycast (OFA) switches, anycast controller

- OFA switches are deployed in the sub-networks of an AS and are interconnected through IP tunnel or dedicated link to form a wide-area layer 2 network

- The anycast controller has IP connections with OFA switches
Design of Anycast Controller

- OpenFlow controller: POX, NOX, Maestro, Beacon, SNAC, Floodlight ...

- POX: Rapid development and prototyping of network control software

- Four modules in Controller to support anycast service
  - Information Gathering Module
  - Routing Decision Module
  - Address Resolution Module
  - Data Transmission Module
Simulation Experiment

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical machine (PM)</td>
<td>2.5GHz Intel Core i5 processor 4GB 1600MHz memory</td>
</tr>
<tr>
<td>Operating system (OS) of PM</td>
<td>Mac OS X 10.7.5 Lion</td>
</tr>
<tr>
<td>Virtual machine (VM)</td>
<td>Oracle VirtualBox 4.2.4 r81684</td>
</tr>
<tr>
<td>VM image</td>
<td>Official Mininet 2.0.0</td>
</tr>
<tr>
<td># of OFA switches</td>
<td>100</td>
</tr>
<tr>
<td># of anycast server</td>
<td>5</td>
</tr>
<tr>
<td>Queue size of OFA switch</td>
<td>64 packets</td>
</tr>
<tr>
<td>Queue size of anycast client</td>
<td>64 packets</td>
</tr>
<tr>
<td>Packet size</td>
<td>64 bytes</td>
</tr>
</tbody>
</table>
Performance Analysis (1)

The (a) anycast request delay and (b) request loss probability predicating by the existing hop-based scheme and proposed load-aware scheme against traffic rate under 1 Mbps bandwidth without background traffic.
The (a) anycast request delay and (b) request loss probability predicated by the existing hop-based scheme and proposed load-aware scheme against request generation rate under 2 Mbps bandwidth and 0.5 Mbps background UDP traffic
Performance Analysis (3)

The (a) anycast request delay and (b) request loss probability predicated by the existing hop-based scheme and proposed load-aware scheme against the three cases with the traffic rate 200 packets/s
Conclusions

• This paper has presented a load-aware anycasting implementation in OpenFlow networks.

• Extensive Mininet experiments have been conducted to validate the effectiveness and accuracy of the proposed scheme.

• The results have demonstrated that the performance of the developed anycasting outperforms that of existing solutions in terms of anycast request delay and loss probability.
Thanks for your attention!