Shadow MACs: Scalable Label-switching for Commodity Ethernet

Kanak Agarwal, Colin Dixon*, Eric Rozner, John Carter
IBM Research, Austin, TX

* now at Brocade
SDN: The Future!

• Rose-colored glasses:
  Fine-grained, dynamic control of the network

• Supported by:
  • Flow mod’s based on diverse set of pkt hdr fields
  • Network measurements obtained in milliseconds\(^1\)
  • Flow mods installed hundreds of times a second\(^2\)

---

SDN: The Future!

- Rose-colored glasses: Fine-grained, dynamic control of the network
- Supported by:
  - Flow mod's based on diverse set of pkt hdr fields
  - Network measurements obtained in milliseconds
  - Flow mods installed hundreds of times a second

Most SDN deployments limited to overlays or small production environments

SDN: The Future?

- Significant issues can arise at scale!

- Flow mods based on diverse set of pkt hdr fields
- TCAMs expensive, only few 1,000 rules supported

- Network measurements obtained in milliseconds
- Flow mods installed hundreds of times a second

Consistent network updates are hard!
Label Switching to the Rescue!

- Label switching common forwarding mechanism (Frame Relay, ATM, MPLS, …)

- We’ll borrow:
  - **Label-switched core**: fixed-width, exact-match lookups map easily into large forwarding tables
  - **Opaque labels**: not assoc to physical endpoint in n/w
Our solution: Shadow MACs

- Opaque forwarding label: Destination MAC address
- Fast, cheap and large fwd’ing tables already in switch!
- OpenFlow flow mods on ingress/egress guide onto paths

1. Ingress switch assigns labels to packets

<table>
<thead>
<tr>
<th>MAC SRC</th>
<th>MAC DST</th>
<th>PORT DST</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>80</td>
<td>B -&gt; B1 out: port</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>*</td>
<td>B -&gt; B2 out: port</td>
</tr>
</tbody>
</table>

2. Core fwd’s on labels

B1 route

B2 route

3. Egress switch rewrites MAC address

<table>
<thead>
<tr>
<th>MAC DST</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B1 -&gt; B out: port</td>
</tr>
<tr>
<td>B2</td>
<td>B2 -&gt; B out: port</td>
</tr>
</tbody>
</table>

B1 route

B2 route

B
Shadow MACs: Rerouting

• **Opaque labels**: no physical host → preinstall routes

• **Ingress guiding**: Changing routes now an atomic action!

1. Controller preinstalls four routes from A to B, each with own shadow MAC address

<table>
<thead>
<tr>
<th>MAC</th>
<th>DST</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B</td>
<td>B1 -&gt; B out: port</td>
</tr>
<tr>
<td>B2</td>
<td>B</td>
<td>B2 -&gt; B out: port</td>
</tr>
<tr>
<td>B3</td>
<td>B</td>
<td>B3 -&gt; B out: port</td>
</tr>
<tr>
<td>B4</td>
<td>B</td>
<td>B4 -&gt; B out: port</td>
</tr>
</tbody>
</table>

2. Controller also preinstalls rewrite rules on egress
Shadow MACs: Rerouting

• **Opaque labels:** no physical host → preinstall routes

• **Ingress guiding:** Changing routes now an atomic action!

1. Controller preinstalls four routes from A to B, each with own shadow MAC address

<table>
<thead>
<tr>
<th>MAC</th>
<th>DST</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B1</td>
<td>-&gt; B</td>
</tr>
<tr>
<td></td>
<td>out: port</td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td>B2</td>
<td>-&gt; B</td>
</tr>
<tr>
<td></td>
<td>out: port</td>
<td></td>
</tr>
<tr>
<td>B3</td>
<td>B3</td>
<td>-&gt; B</td>
</tr>
<tr>
<td></td>
<td>out: port</td>
<td></td>
</tr>
<tr>
<td>B4</td>
<td>B4</td>
<td>-&gt; B</td>
</tr>
<tr>
<td></td>
<td>out: port</td>
<td></td>
</tr>
</tbody>
</table>

2. Controller also preinstalls rewrite rules on egress
Shadow MACs: Rerouting

- **Opaque labels**: no physical host → preinstall routes
- **Ingress guiding**: Changing routes now an atomic action!

1. Single flow mod to ingress switch
   
   switches paths

2. Traffic immediately switches to green route
Benefits

• Controller guides pkts onto intelligently selected paths
  • Load balancing, link fail-over, route via middleboxes, differentiated services, …

• Decouples network edge from core
  • Consistent n/w updates, fast rerouting, multi-pathing, …

• Maps fine-grained matching to fixed destination-based rules
  • Pushes TCAM rules to FDB, limits TCAM usage in core

• Implementable today!
TCAM Usage

• TCAM usage:
  • Core switches use little/no TCAM rules
  • TCAM rules limited to edges, best case (OVS) uses no TCAM
  • L2 forwarding tables are typically largest tables in switches
  • Scales better (up to 124x more L2 entries than TCAM)

<table>
<thead>
<tr>
<th></th>
<th>Broadcom Trident</th>
<th>IBM Rackswitch</th>
<th>HP ProVision</th>
<th>Intel FM6000</th>
<th>Mellanox SwitchX</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCAM</td>
<td>~4K</td>
<td>1K</td>
<td>1500</td>
<td>24K</td>
<td>0?</td>
</tr>
<tr>
<td>L2/Eth</td>
<td>~100K</td>
<td>~124K</td>
<td>~64K</td>
<td>64K</td>
<td>48K</td>
</tr>
<tr>
<td>X more</td>
<td>~25x</td>
<td>~124x</td>
<td>~42x</td>
<td>~2.6x</td>
<td>∞</td>
</tr>
<tr>
<td>L2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10Gbps Ethernet Switch Table Sizes (# entries) [1]
Fast, Consistent Updates

- Consistent Route updates:
  - SDN controller can pre-install routes
  - **Atomic reroute**: single flow-mod at ingress switch
- Two ways to achieve:
  - MAC address rewriting (OpenFlow)
  - ARP spoof (SDN controller sends GARP response)
E2E Multi-pathing

• SDN controller can allocate multiple distinct paths (shadow MACs) per destination

• OVS can allocate flows in round-robin fashion

• Benefits over ECMP
  • True L2 solution (ECMP is L3)
  • More control: per-path, instead of per-hop
**Testbed Methodology**

- UDP pkts start on Route 1, switch to Route 2

- **Goal:** measure # times per-pkt consistency violated, compare:
  - Shadow MAC rerouting
  - Traditional, iterative OpenFlow (order: sw4, sw2, sw1)
    - Uses Static Flow Pusher (barrier msg’s not implemented)
Per-Pkt Consistency

- CDF over 700 runs: at least 1 pkt misrouted every time
- Loss in ~5% of cases
- ShadowMACs: no inconsistency & no loss!

Figure 3 plots a CDF of the number of incorrectly routed packets. The graph shows that traditional solutions (Iterative OpenFlow rerouting) have a high probability of misrouted packets, whereas ShadowMAC rerouting has a much lower probability of misrouted packets, with a CDF that approaches 1.0.

Figure 4: A CDF of the time to install a flow mod using Openflow barrier messages. The graph illustrates the efficiency of ShadowMAC rerouting compared to traditional methods, with a significantly reduced time to install a flow mod.
Iterative Flowmod Overhead

- Iterative schemes pay per-switch overhead
- Shadow MAC overhead only at single switch
- 20-40 ms faster than traditional schemes
Related Work

• Have we seen this before?

  • Label-switching common

  • Motivated by separate, clean host-network, operator-network and packet-switch interfaces

  • MPLS: Little support in switches

  • Consistent route updates [Reitblatt12, Jin14, …]
Summary

• SDN networks have issues at scale
  • Dynamic, fine-grained control of the network is challenging

• Label-switching using Shadow MACs is promising
  • Flexible edge steers traffic via OVS
  • Opaque labels (destination MAC) allow pre-installation of routes
  • Very practical: DMAC tables are widespread, large and fast

• Shadow MACs is a flexible architecture
  • Enable fast, atomic route updates, straight-forward mechanisms to implement multi-path, differentiated services, load-balancing, etc
Questions?

• Eric Rozner  
erozner@us.ibm.com

• Co-authors:  
Kanak Agarwal, Colin Dixon, John Carter

We are hiring at IBM Research in Austin!  
• All areas  
• All experience-levels