Hands-on Labs Presentation

P4 Bootcamp 2015
2 different P4 exercises

- implementing EasyRoute, a custom source routing protocol
- realizing TCP flowlet switching
Pb1 - EasyRoute (source routing)

- very simple header

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Num hops</th>
<th>Port_1</th>
<th>Port_2</th>
<th>...</th>
<th>Port_N</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>8B of 0s</td>
<td>4B</td>
<td>1B</td>
<td>1B</td>
<td>...</td>
<td>1B</td>
<td></td>
</tr>
</tbody>
</table>

- preamble lets you identify EasyRoute packets
- do not have to worry about encapsulation / decapsulation at end hosts
- at each hop:
  - use the 1st port number as the outgoing port
  - decrement the number of hops
  - pop the head of the list
Pb1 - EasyRoute (source routing)

- let’s send ‘Hello’ from h1 to h3

- when it leaves h1:
  00000000 00000000 | 00000002 | 03 | 01 | Hello

- when it leaves sw1 (on port 3):
  00000000 00000000 | 00000001 | 01 | Hello

- when it leaves sw3 (on port 1):
  00000000 00000000 | 00000000 | Hello
Pb2 - TCP Flowlet Switching

- leverage the burstiness of long TCP flows to achieve more accurate load balancing
- we start with regular ECMP, then add a flowlet_id to the list of hash fields used to select the next hop
- the flowlet_id is incremented everytime we observed a gap > 50ms between packets
Pb2 - TCP Flowlet Switching

- \text{crc16(5-tuple)} \rightarrow \text{flow\_idx}
- \text{register1[flow\_idx]} \rightarrow \text{last\_timestamp}
  \text{register2[flow\_idx]} \rightarrow \text{flowlet\_id}
- \text{if now - last\_timestamp} > 50\text{ms}: \text{flowlet\_id++}
- \text{crc16(5-tuple + flowlet\_id)} \rightarrow \text{ecmp\_nhop}
Debugging your P4

- use the p4-validate tool to check that your program is correct:
  - p4-validate p4src/source_routing.p4
- look at the pcap files (one for each port)
- look at the switch logs in /tmp/
- ask us :)
