Simplifying Software-Defined Network Optimization Using SOL

Victor Heorhiadi
UNC Chapel Hill

Michael K. Reiter
UNC Chapel Hill

Vyas Sekar
Carnegie Mellon University
Networks are Critical Infrastructure

• Leisure services
• Record keeping
• “Cyber-physical” systems
• Internet of things
Security and Resiliency

• Achieved by integrating with the network goals and applications
A Traffic Engineering Problem

1. Formulate the problem
2. Solve the optimization
3. Parse the output
4. Configure every switch

- Satisfy demands C1, C2 100%; minimize max link utilization
- C1: N1→N5 1 Gbps
- C2: N2→N5 100 Mbps
Service Chaining

Application has

1. Policy
2. Optimization

<table>
<thead>
<tr>
<th>C1: N1 → N5</th>
<th>Web, 100 Mbps</th>
<th>FW → Proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2: N1 → N5</td>
<td>Other, 50 Mbps</td>
<td>FW → IPS</td>
</tr>
<tr>
<td>C3: N2 → N5</td>
<td>Web, 100 Mbps</td>
<td>Proxy</td>
</tr>
<tr>
<td>C4: N2 → N5</td>
<td>Other, 100 Mbps</td>
<td>IPS</td>
</tr>
</tbody>
</table>

Satisfy policy routes; do not overload middleboxes
Motivation

• Networking applications are often optimizations
  • Maxflow, shortest path, traffic engineering
  • SIMPLE, Hedera, ElasticTree, Panopticon, SWAN…
• (Re)inventing the wheel
• Need for a common abstraction layer

1. Formulate the problem
2. Solve the optimization
3. Parse the output
4. Be sad 1-3 takes too long
5. Develop a heuristic
6. Configure every switch
Challenges

• Generality: capturing a variety of applications
• Efficiency: reacting fast to changing demands/policies
• Both!
SOL: SDN Optimization Layer

SDN Applications
- e.g., Service chaining, Traffic engineering

SOL API

SOL

Network data

Network routes

Solver, e.g., CPLEX

Control platform, e.g., OpenDaylight

Diverse set

Logically centralized
Insight: Path Abstraction

- Policies are path predicates
- Resource constraints are easily computed
- Suited for path-based constraints (TCAM)
Path Challenge

- Exponential number of paths
SOL Process

- Path selection
  - Includes path filtering (based on predicates)
  - Acts as a heuristic

- Optimization
  - Vol on $p_1$

- Rule generation
  - Rules for $p_1$
Further opportunities

• Composing applications

• Layer of common “add-on” features