SYNTHESIZING NETWORK UPDATES AND CONTROLLERS

MADHUSUDAN PARTHASARATHY
SHAMBWADITYA SAHA

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

IN CONSULTATION WITH:

NETWORKING GROUP@ILLINOIS (GODFREY, CEASAR)
NETWORKING GROUP@PENN (BOON)

NSF EXCAPE
Why synthesize networks:

• Networks are undergoing a renaissance.
  - SDNs, OpenFlow, Programming Languages (Frenetic)

• Small enough domain where synthesis can be effective
  - 1000s of nodes with routing tables
  - Tractable for verification (Veriflow @ Illinois)
  - Synthesis can solve real-sized problems in this domain.
NETWORKS AND SDN

- Packet headers: sequence of bits (large: typically 128/256)
- Nodes / switches: a set $N$
- Ports: places where packets get forwarded from/to
  Two special ports: DROP and WORLD
- Topology $\subseteq (\text{Ports} \times \text{Ports})$
- Forwarding logic at a node:
  Maps $(\text{packet}, \text{port}) \mapsto \text{set of ports}$
- Network:
  $(T: \text{Topology}, \{\text{FwdLogic}@n \mid n \in N\})$

Software-defined networks (SDN):
- Centralized controller
- Openflow interface: controls packet forwarding rules, priorities, timeouts, etc.
Switching logic at a node is given using a large if-then-else rule:

- if (packet matches pattern r1) then (send it to n4)
- else if (packet matches pattern r2) then (send it to n9)
- else if ...
- else ...

Switch can be changed by controller.

Controller maintains high level policies by exerting low-level control:

- routing, access control policies, security,
- traffic load balancing, firewalls, etc.
THE NETWORK UPDATE PROBLEM

Problem 1:
- Current network: N
- Move to a new network \( N' \) that satisfies certain properties
  (e.g., rerouting packets to different destination, routing packets through firewalls, traffic/server load balancing, VM migration, etc.)

Problem 2:
- Synthesize the controller software itself so that the network maintains a property.
The Network Update Problem

Problem 1:
- **Current network**: N
- **Move to a new network** $N'$ that satisfies certain properties (e.g., rerouting packets to different destinations, routing packets through firewalls, traffic/server load balancing, VM migration, etc.)

Given $k$, find $k$ changes to the routing in $N$ to synthesize $N'$
- Bounding $k$ is important to get acceptable synthesized networks as the synthesis problem is underspecified.

E.g., $k=5$; reroute packets that match pattern $\pi$ to a new destination $t$
KEY CHALLENGES

• **Packet explosion**
  Number of nodes, edges are manageable
  Number of packets → LARGE
  **Our solution: abstraction to packet classes**
  (similar to Veriflow, but more involved and accurate)

• **Reachability / recursive properties**
  Inexpressiveness of FOL for stating reachability on graphs
  **Our solution: higher-order logic but encoded using abstractions, uninterpreted functions, etc.**
  (experiences from verifying data-structures in s/w !!)
Compute packet abstractions: PACKET CLASSES

C: Packets that take same path through network BUT ALSO IN NEW NETWORK
Efficient and precise symbolic algorithm

Construct abstract network on packet classes

Recursive definitions modeled using uninterpreted functions and arrays with constraints;
Intuition: Data-structure verification
Scales reasonably well.

Synthesize new routing using SMT constraints

Synthesize forwarding rules
### PRELIMINARY UNOPTIMIZED RESULTS

A data-center topology with ~ 1200 nodes; 4000 rules, Spec: Rerouting packets to new destination

<table>
<thead>
<tr>
<th>K</th>
<th>Sliced Nodes</th>
<th># Packet classes</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>70</td>
<td>700</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>1400</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>130</td>
<td>1650</td>
<td>78</td>
</tr>
<tr>
<td>6</td>
<td>300</td>
<td>3000</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>130</td>
<td>1650</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>130</td>
<td>1650</td>
<td>55</td>
</tr>
<tr>
<td>5</td>
<td>130</td>
<td>1650</td>
<td>67</td>
</tr>
<tr>
<td>6</td>
<td>130</td>
<td>1650</td>
<td>78</td>
</tr>
<tr>
<td>10</td>
<td>130</td>
<td>1650</td>
<td>135</td>
</tr>
</tbody>
</table>
CURRENT / FUTURE WORK

• Robust implementation and scaling experiments

• Are SMT solvers really necessary? Clever algms?
  (AntEater $\rightarrow$ VeriFlow at Illinois)

• Logic for desired properties of networks
  - NDLog (Boon)

• Network controller algorithm synthesis
Controller $\leftrightarrow$ Synthesizer
  Sygus-based solutions?