A Synthesis Approach Towards Automated Management in Software-defined Networks

Anduo Wang, Ufuk Topcu
Boon Thau Loo, Andre Scedrov
University of Pennsylvania
Network Management

- Current computer networks
  - Network fabrics become more powerful
  - Emerging applications demand more reliable and complex services

- Network management
  - Manual, low-level, unpredictable process
  - Silent failures and hidden dependencies common
  - **Software-defined network (SDN)** makes worse
    - Forces management at an individual switch level
  - Emerging applications (cloud, datacenter)
    - Sheer size makes management intimidating

- Lacking rigorous and scalable management
Synthesis Approach

- Network management = Implement control logic on data-plane + construct control logic
- Synthesize provably correct solutions for two families of control logic construction problems
  - Static control logic construction (SCC)
    - Find a control logic satisfying network-wide requirement & invariants
    - Solve as reachability problem
  - Dynamic control logic construction (DCC)
    - Find a strategy that updates control logic in response to network state change
    - Solve as two-player, temporal logic game
  - Automated through model checking and game solvers (e.g., \textit{nuSMV}, \textit{JTLV}, \textit{TuLiP},...)

3
Static Control Logic Construction

Find a control logic satisfying network-wide requirement & priori invariants

<table>
<thead>
<tr>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Forward F₁, Forward F₂, Forward F₃</td>
</tr>
<tr>
<td>F₁</td>
<td>SSH * Deny Allow</td>
</tr>
<tr>
<td>F₂</td>
<td>* Allow</td>
</tr>
<tr>
<td>F₃</td>
<td>* Allow</td>
</tr>
</tbody>
</table>

Configuration 1

<table>
<thead>
<tr>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Forward F₁, Forward F₂, Forward F₃</td>
</tr>
<tr>
<td>F₁</td>
<td>SSH * Deny Allow</td>
</tr>
<tr>
<td>F₂</td>
<td>SSH * Deny Allow</td>
</tr>
<tr>
<td>F₃</td>
<td>* Allow</td>
</tr>
</tbody>
</table>

Configuration 2

[Sigcomm’12] Abstractions for network update, Mark Reitblatt et, al.

- Requirement: given configuration 1, set it to 2
- Priori invariant: deny untrusted/SSH traffic, allow other
- Synthesizing solution
  - Solve as reachability problem in model checker nuSMV
  - Output: an ordering of rule updates
Dynamic Control Logic Construction

Find a strategy that updates control logic in response to network state change

Flows: U, G, S, F

- F: Forward to F
- S: Forward to F
- U: Forward to F
- G: Forward to F

Load Balancing

Switches: I (ingress), F₁, F₂ (for two servers)
Flows: U (untrusted), G (guest), S (student), F (faculty)

Network change: routing paths
Pirori invariant: deny U flows

- Solve as two-player temporal logic game
  - Control logic = routing path rule (environment player) + access control rule (system player)
- Find a winning strategy for access-control rules against all path changes
  - Winning strategy: invariant preserving
Dynamic Control Logic Construction

- **Input** (Using *TuLiP* for interfacing game solver)
  - Two player variables: routing-path (IF), access-control (F1ac, F2ac, lac)
  - Invariant: Always deny untrusted flows
- **Output**: a strategy with finite memory

Flows: $U,G,S,F$

**Solution**

- $0$: F1ac: Deny, F2ac: Deny, lac: Deny, IF: 1
- $1$: F1ac: Deny, F2ac: Allow, lac: Allow, IF: 1
- $2$: F1ac: Deny, F2ac: Deny, lac: Allow, IF: 2

*F*: Forward to $F_2$
*S*: Forward to $F_1$
*U*: Forward to $F_1$
*G*: Forward to $F_1$

*U*: Monitor SSH
*F*: Allow all
*U*: Deny other
*G*: Monitor SSH
*S*: Deny other
*S*: Allow all
Scaling by Abstraction

1 + 12 = 13 variables, larger search space

A larger dynamic control logic construction problem

1 environment variable + 3 system variables

Topology-based abstraction by grouping nodes
Scaling by Abstraction

1 + 12 = 13 variables, larger search space

A larger dynamic control logic construction problem

Synthesize solution for the abstract problem
Scaling by Abstraction

IF: 2, Iac: Allow?

Iac1: Allow
Iac2: Allow
Iac3: Allow
Iac4: Allow

Implementation

F1ac: Deny?
F2ac: Deny?

Synthesize solution for the abstract problem
Conclusion

- Contributions
  - Formalize two families of network management problems
  - Synthesize provably-correct control logic
  - Investigate abstraction technique for scaling

- Future Work
  - Distributed controllers in SDN
    - Contract-based synthesis, contract discovery
  - Virtual network – killer application in SDN
    - Bi-simulation framework