Formal Synthesis in Software-Defined Networks

Anduo Wang
University of Pennsylvania

A Project from the NSF Expeditions on Software Synthesis

http://excape.cis.upenn.edu
Management Challenges

Enforce policy

Maintain network-wide invariants

Ensure isolation
Allocate resources

Network Migration

Network Virtualization

Alice’s Controller

Bob’s Controller

Openflow Switch

Translator

Openflow Switch
Management Solutions

Mechanism for Consistent updates

Many heuristics Optimization algorithms

Flowvisor Emerging…

Network Migration

Alice’s Controller  Bob’s Controller

Openflow Switch

Network Virtualization

Alice’s Controller  Bob’s Controller

Translator

Openflow Switch
Formal Synthesis Approach

Traditional
- (Very) High initial investment
- Ad-hoc solution for each management problem

Formal synthesis
- Provably-correct general framework
- Automated solution for small instance, compositional synthesis for scaling
- Guidance for quickly developing heuristics on large networks
Formal Synthesis Approach

Closed system synthesis
Reachability problem solved by model checking

Open system synthesis
Compositional synthesis

(Ongoing)
Closed System Synthesis for Migration

• Formulate network migration as reachability problem
  – Model network migration by a transition system
  – Find a migration ordering $t$ (a sequence of atomic updates) from initial network state to the target final state s.t. constraints $P$ holds during all transient states along $t$
  – A migration $t$ exists if $\neg P$ does not always hold

• Solving by model checker
  – Model check $\neg P$ on the transition system
  – Counter example of $\neg P$ gives $t$
Solution for VM Migration

- Specification:\[ F((v2 = s4 \& v4 = s5) \& v5 = s7) \& G((v2 != v4 \& v2 != v5) \& G\ bandwidth13 < 2)\] is false.

- As demonstrated by the following trace:

  Trace Description: LTL Counterexample
  Trace Type: Counterexample
  -> State: 1.1 <-
  \[ v2 = s2 \]
  \[ v4 = s7 \]
  \[ v5 = s8 \]
  \[ bandwidth13 = 1 \]
  -> State: 1.2 <-
  \[ v4 = s5 \]
  -> State: 1.3 <-
  \[ v2 = s4 \]
  -- Loop starts here
  -> State: 1.4 <-
  \[ v5 = s7 \]
  -- Loop starts here

- [HotSDN’12] Walk the Line: Consistent Network Updates with Bandwidth Guarantees

- Migration goal
  - Move V5, V4, V2 to S7, S5, S4.

- Constraints P
  - One substrate node can hold only one VM
  - Heavy dashed lines show inter-VM communications.

- Solution
  - Migrating with sequence V4, V2, V5 succeeds to migrate all nodes while migration with sequence V5, V2, V4 can migrate only one node.
Solution for Configuration Migration

- **Migration goal**
  - Configuration I -> Configuration II

- **Constraints P**
  - Enforce a security policy that denies SSH traffic from untrustworthy hosts, but allows all other traffic to pass through the network unmodified

- **Solution**
  - Update I to forward S traffic to F3
  - Update F2 to deny SSH packets
  - Update I to forward G traffic to F2
Formal Synthesis Approach

Open system synthesis
Compositional synthesis

(Ongoing work)

Closed system synthesis
Rechability problem solved by model checking
Open System Synthesis for Virtualization
Compositional Synthesis for Virtualization
Conclusion: Formal Synthesis

Closed system synthesis
Reachability problem solved by model checking

(Ongoing work)
Open system synthesis
Compositional synthesis