Enterprise Network Policies

Cloud Network
Tenant Network Policies

Reachability:
A can talk to B

Waypoints:
C to B traffic goes through a Firewall
Cloud Network Policies

Network isolation:
Tenant 1 and 2’s traffic must not affect each other

Network resource management

Fault tolerance
Intent-based networking

High-level language to specify policies → Synthesize → Policy-compliant network configurations

INPUT

OUTPUT
GENESIS
SYNTHESIZING FORWARDING TABLES IN MULTI-TENANT NETWORKS

[Subramanian, D'Antoni, Akella, POPL17]
Software-defined Networks

Enforcing policies using conventional distributed networks is difficult

Programmable switch rules:

**Match:** Packet headers

**Action:** Forward to next switch

SSH traffic at $S_3$ is forwarded to $S_7$
High-level language to specify policies

Support for complex and diverse policies

Genesis

Switch forwarding tables

Enforcing certain policies is NP-complete

Genesis uses Satisfiability Modulo Theories (SMT) solvers to synthesize forwarding tables
Outline of the Talk

• Motivation
• Synthesis of forwarding tables in Genesis
• Scaling to large workloads: Tactics
• Genesis extensions and conclusions
Synthesis Approach

High-level policies + Topology

Constraints on \textit{Fwd} and \textit{Reach}

\textbf{(Fwd, Reach)}

Paths from \textit{Fwd} and \textit{Reach} solution

Forwarding tables

Abstract Representation

INPUT

OUTPUT
Semantics of \((\text{Fwd}, \text{Reach})\)

\[ \text{Fwd}(S_1, \text{ID}) = S_2: \text{Switch } S_1 \text{ forwards to } S_2 \]

\[ \text{Reach}(S_2, \text{ID}) = 1: \text{Specifies that } S_2 \text{ is reachable in 1 step from source} \]
Reachability Constraints

If a switch is reachable in $k$ steps, one of its neighbors must be reachable in $k - 1$ steps.
**Policy Constraints**

**Waypoint:** Blue Tenant specifies path must traverse through $S_4$

**Isolation:** Blue Tenant and Red Tenant paths do not share any link

**Traffic Engineering:** Using SMT-OPT

![Diagram showing network nodes and links with reachability and ID labels]
THE END?
Baseline Synthesis Evaluation Setup

- Genesis implemented in Python, uses Z3 SMT solver
- Multi-tenant isolation: Each tenant has a single reachability policy, and all tenant paths are mutually isolated
- Medium-sized fat-tree datacenter topologies
Baseline Synthesis Evaluation

To scale to large networks and workloads, we need to further algorithmic insights and optimizations.

Synthesis time for over 60 tenants takes >5000s
SCALING TO LARGE WORKLOADS

TACTICS
Tactics: Motivation

Edge-to-edge paths: 272
Large search space

Use network structure to specify path properties
Tactics as regular expressions

No Edge Tactic:
\( \not (\text{Edge} \,* \, \text{Edge} \,* \, \text{Edge}) \)
Genesis uses tactics as a search strategy to *eliminate* constraints.

No Edge Tactic ensures no intermediate edge switch.

- \( \text{Reach}(C_1) = k - 1 \)
- \( \text{Reach}(A_1) = k - 1 \)
- \( \text{Reach}(E_1) = k - 1 \)

\( \text{Reach}(S) = k \)
Tactics: Algorithmic Properties

- Specified using a restricted subset of regular expressions
- **Sound** and **Complete** algorithm for enforcing them
- Policy-agnostic
- The operator can develop a repository of tactics based on their topology
Tactics: Evaluation

Multi-tenant isolation workload

Valley-Free Tactic and No Edge Tactic

Valley-Free Tactic speedup: 400x
Outline of the Talk

• Motivation
• Synthesis of forwarding tables in Genesis
• Scaling Genesis: Tactics and Divide-and-Conquer
• Genesis extensions and conclusions
Genesis Extensions
Network Resilience

Single path: Not resilient

Cloud network

$t$-resilience: For events under $t$ arbitrary link failures, there exists a valid path
Policy-compliant Resiliency

For 1-resilience, backup path must be edge-disjoint from original path

Sound transformation of input policies to provide \( t\)-resilience
**Minimal Reactive Network Repair**

**Best repair**: Minimize change overhead

*Genesis uses MaxSMT*
Network Repair Evaluation

Multi-tenant isolation workload

One switch-failure, network repair such that number of switches affected is minimized

For larger workloads, repair is faster than re-synthesis.
CONCLUSION
High-level policies on paths and switches

Genesis

Switch forwarding tables satisfying policies

OSPF and BGP configurations

Efficient optimal repair