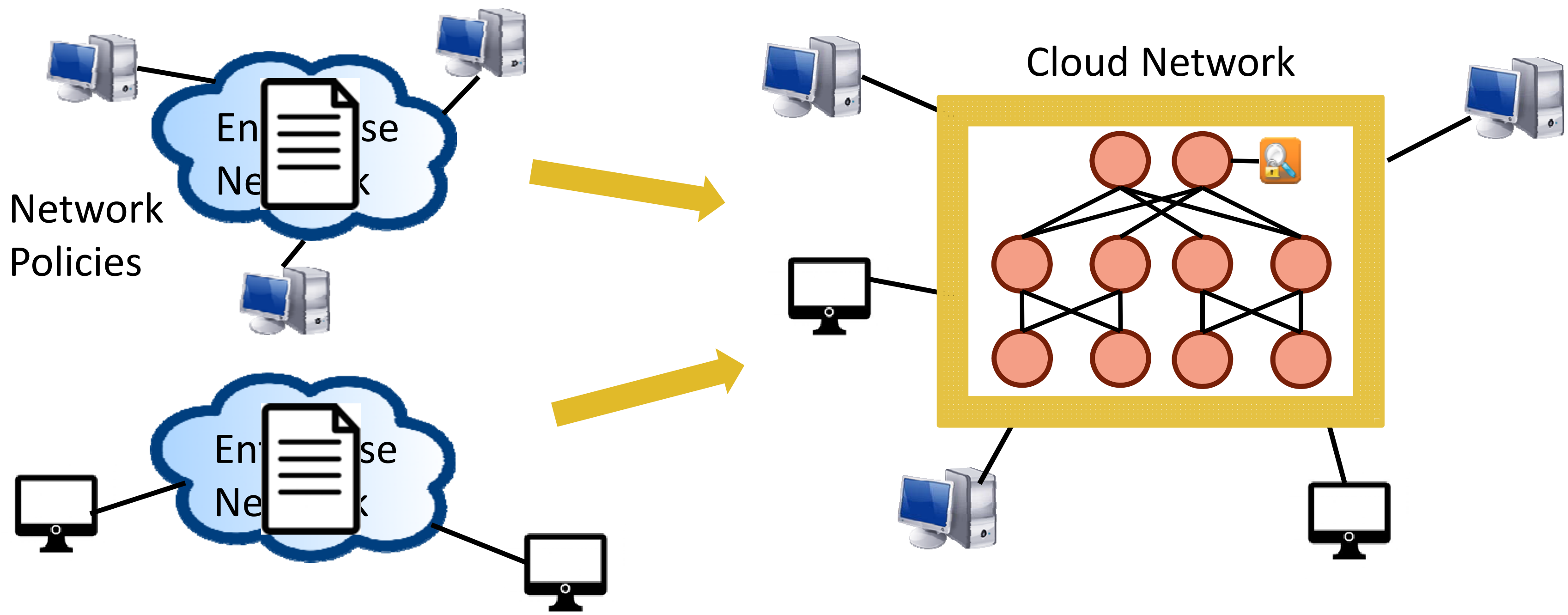
FOUNDATIONS OF INTENT-BASED NETWORKING

Loris D'Antoni

Aditya Akella

Aaron Gember-Jacobson

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
madPL

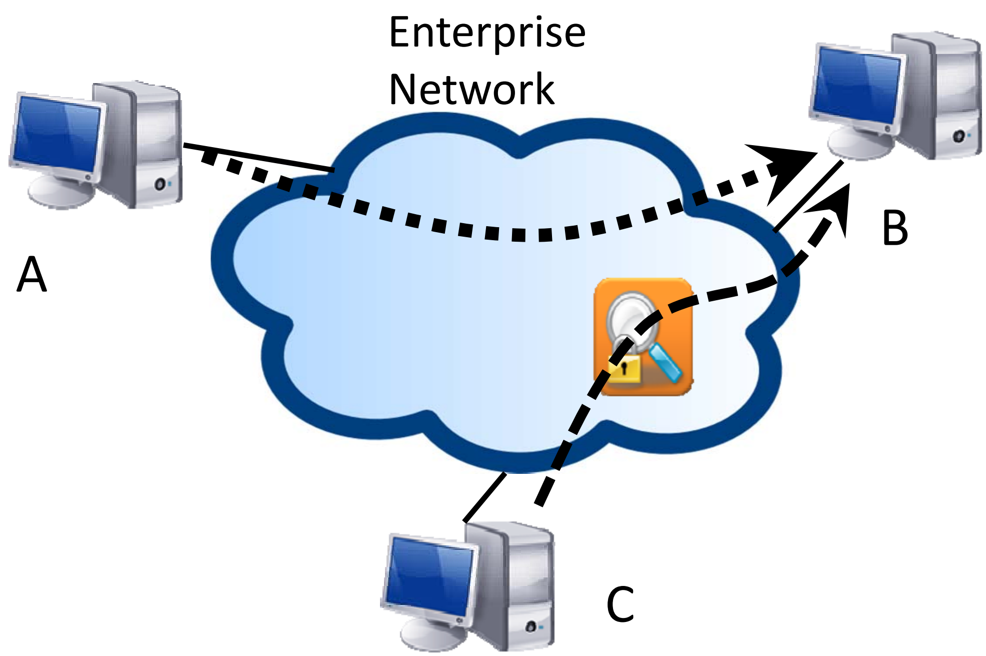


Network Policies
Cloud Network

Tenant Network Policies

Reachability:
A can talk to B

Waypoints:
C to B traffic goes
through a Firewall 



Cloud Network Policies

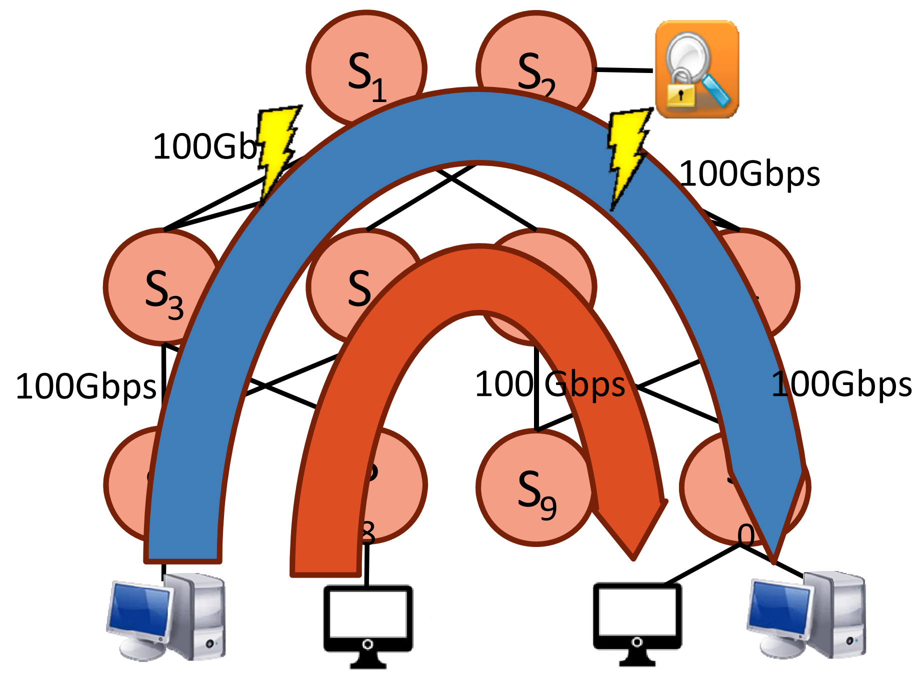
Tenant 1 Network Policies

Tenant 2 Network Policies

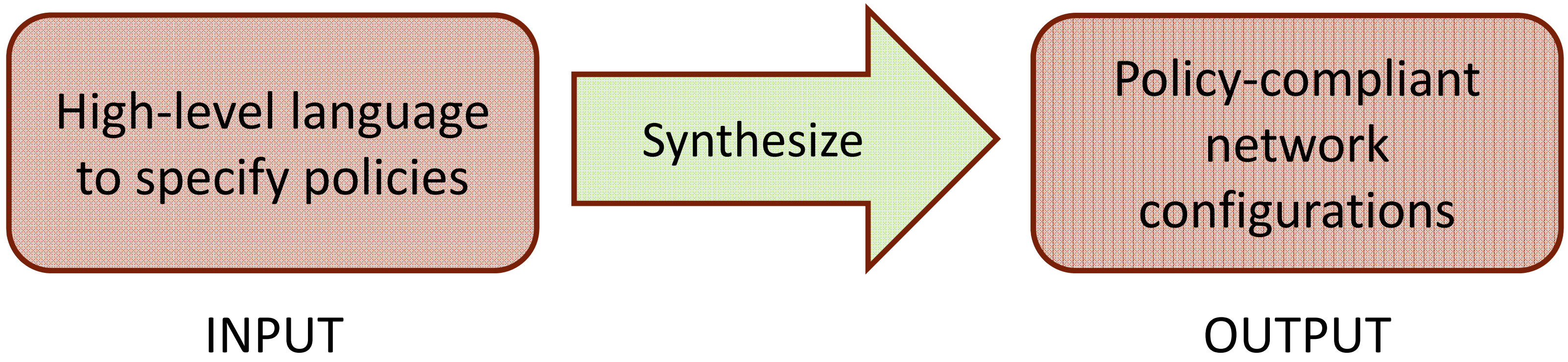
Network isolation:
Tenant 1 and 2's traffic must not affect each other

Network resource management

Fault tolerance



Intent-based networking



Kausik Subramanian

GENESIS

SYNTHESIZING FORWARDING TABLES IN MULTI-TENANT NETWORKS

[Subramanian, D'Antoni, Akella, POPL17]

Software-defined Networks

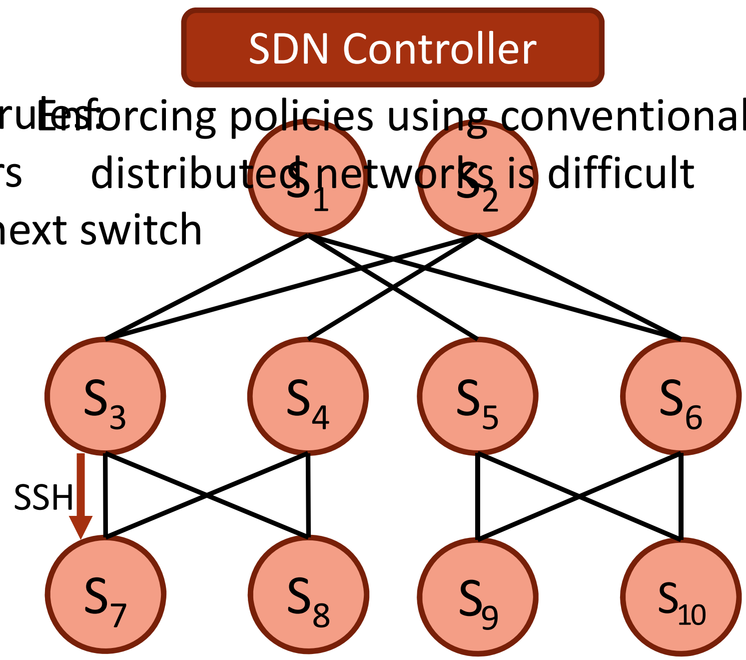
Programmable switch rules:

`Match`: Packet headers

`Action`: Forward to next switch

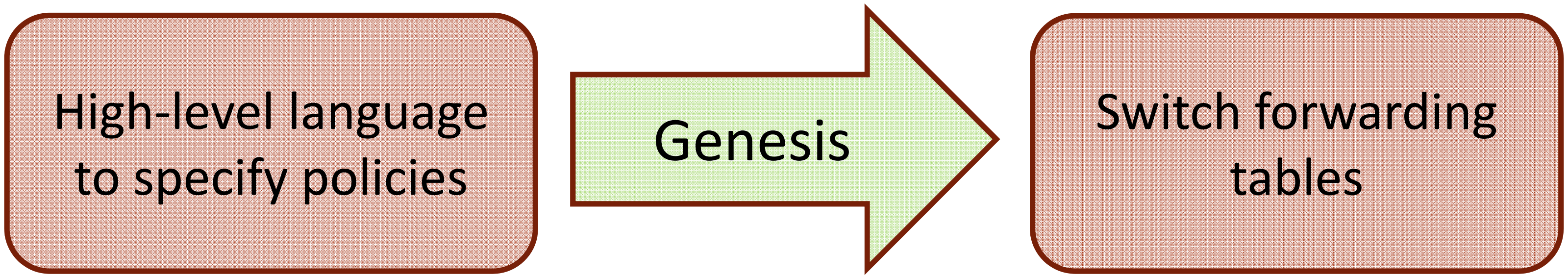
Enforcing policies using conventional distributed networks is difficult

Centralized controller enforces policies



SSH traffic at S_3 is forwarded to S_7

Support for complex
and diverse policies



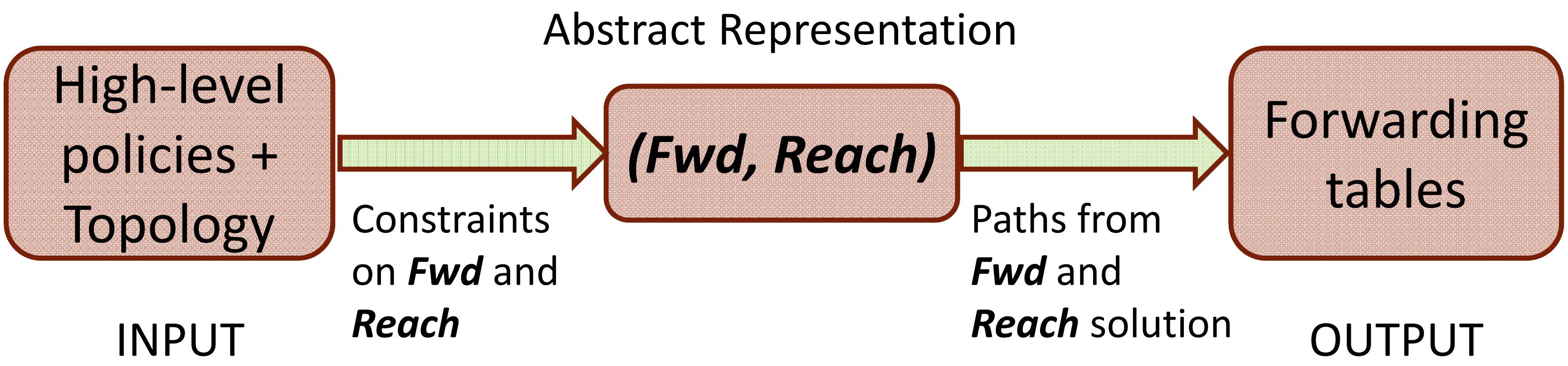
Enforcing certain policies
is NP-complete

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

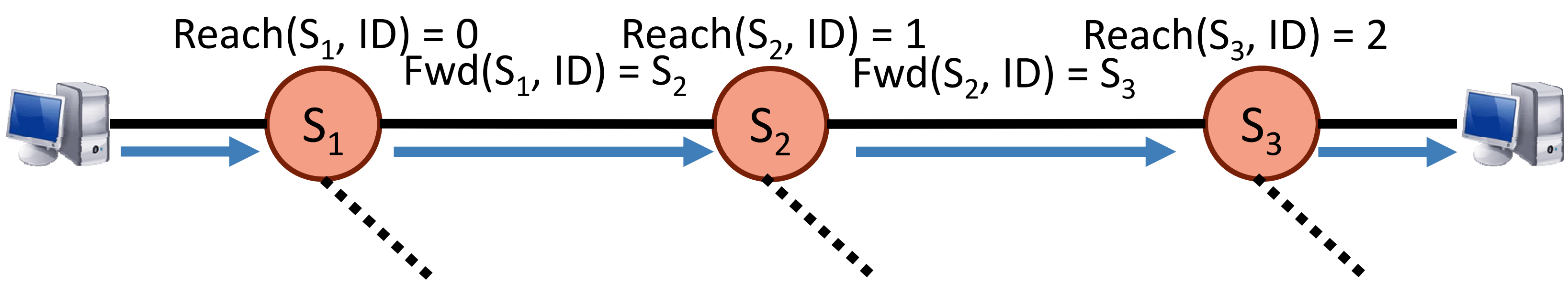
Synthesis Approach



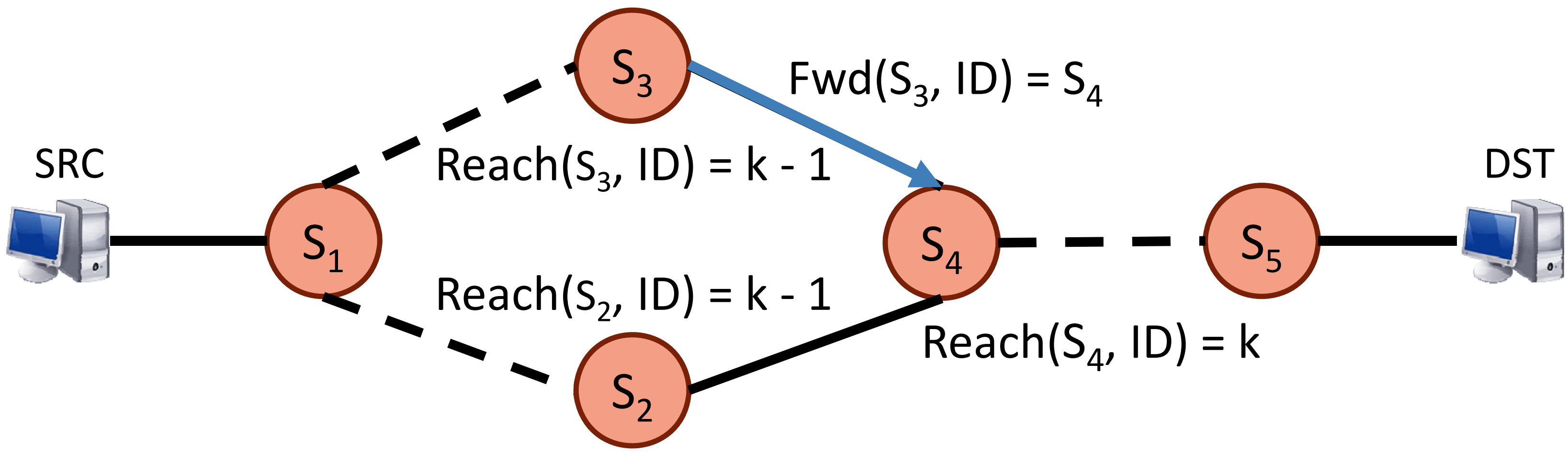
Semantics of (Fwd, Reach)

Fwd(S_1, ID) = S_2: Switch S_1 forwards to S_2

Reach(S_2, ID) = 1: Specifies that S_2 is reachable in 1 step from source



Reachability Constraints



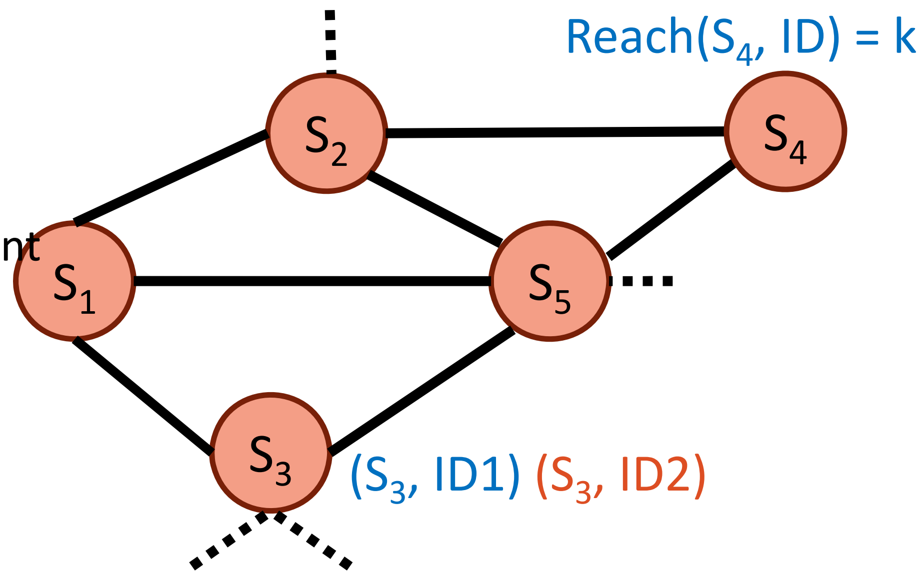
If a switch is reachable in k steps,
one of its neighbors must be reachable in k - 1 steps

Policy Constraints

Waypoint: Blue Tenant specifies path must traverse through S_4

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

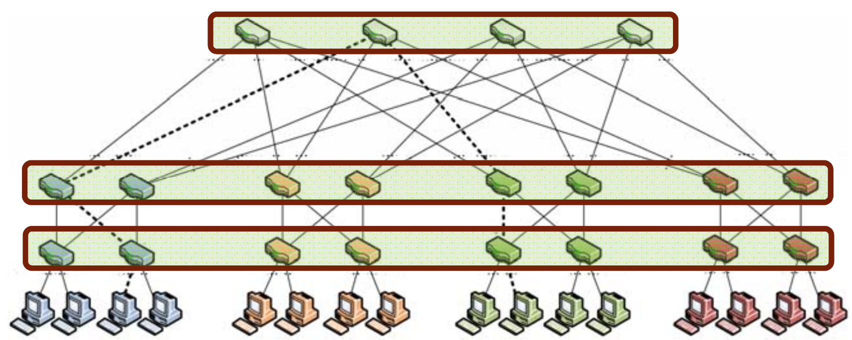
Traffic Engineering: Using SMT-OPT



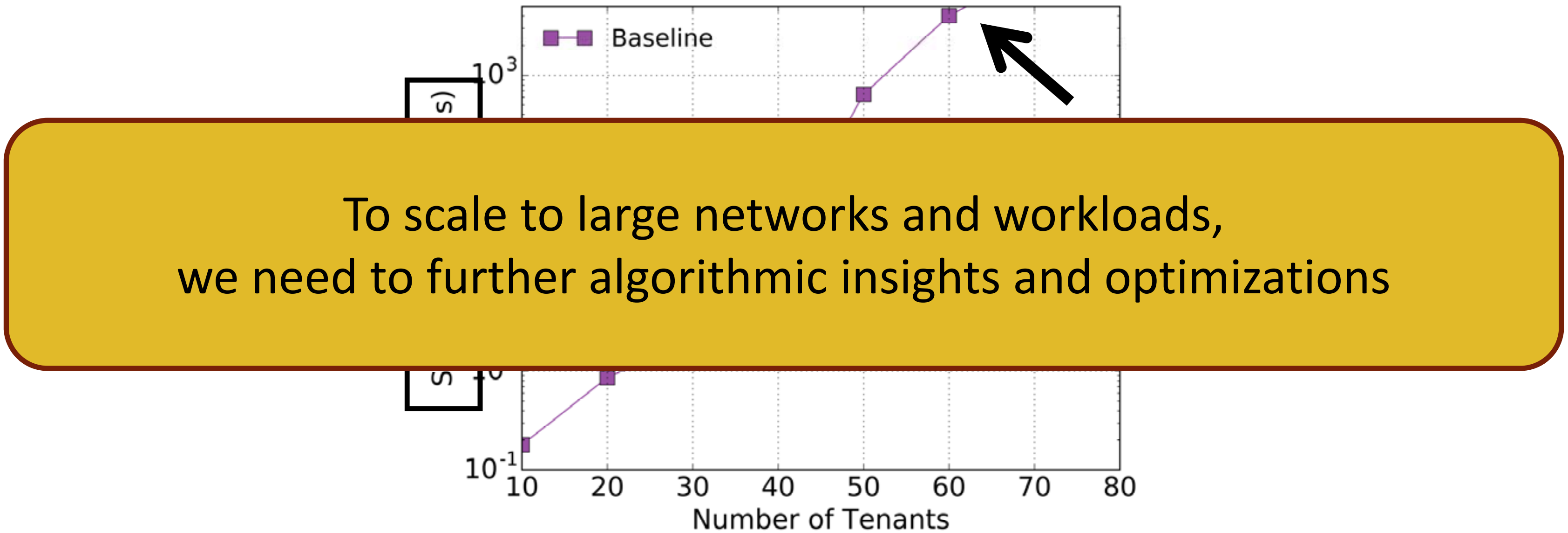
THE END?

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



Synthesis time for over 60 tenants takes >5000s

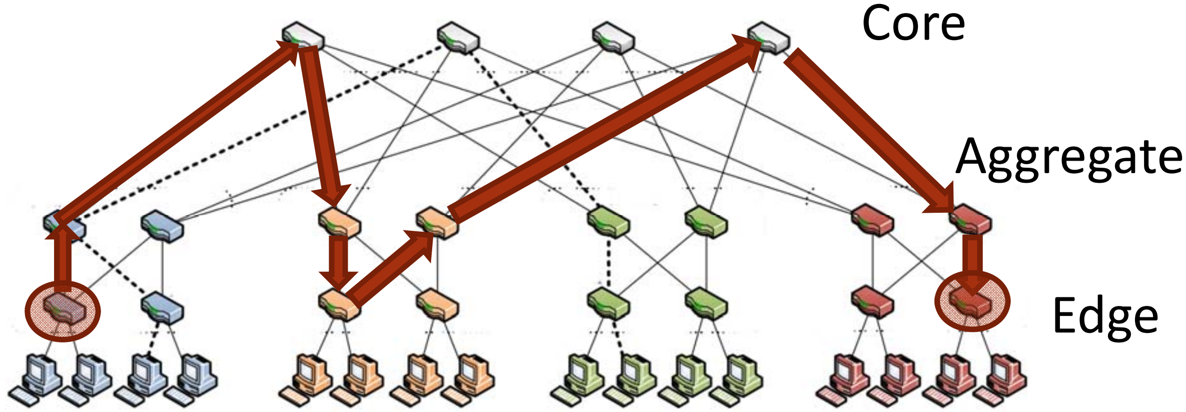
SCALING TO LARGE WORKLOADS

TACTICS

Tactics: Motivation

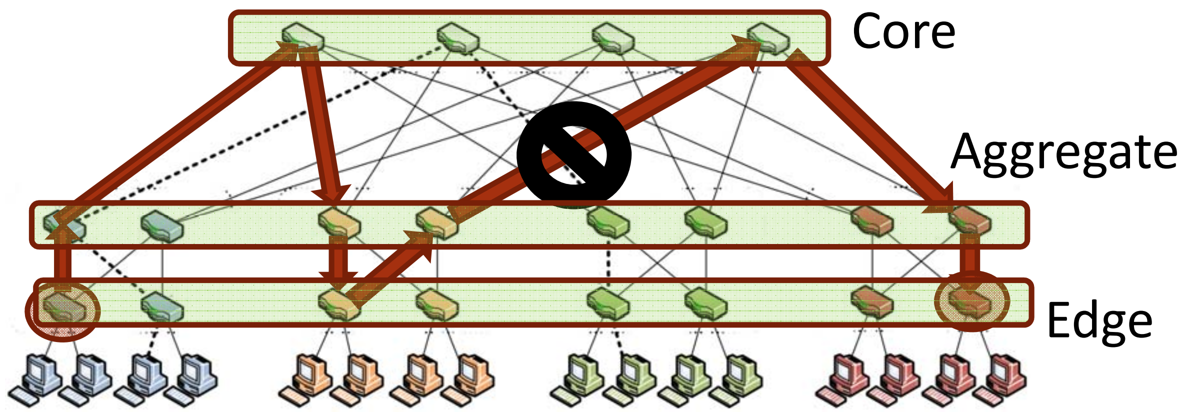
Edge-to-edge paths: 272
Large search space

Use network structure to specify path properties



Tactics as regular expressions

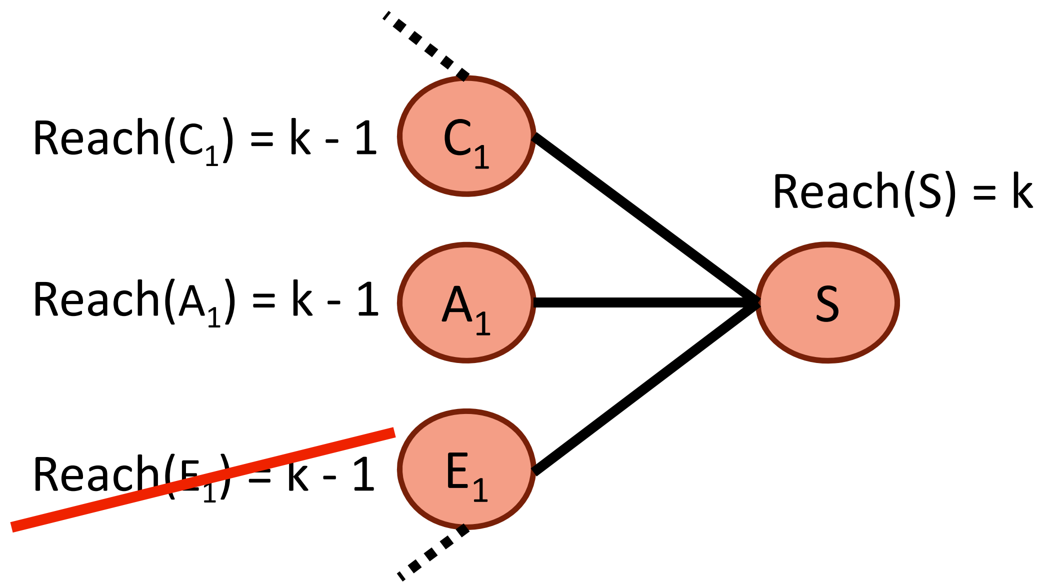
No Edge Tactic:
Not (Edge . Edge .* Edge)*



Tactics: Constraint Reduction

Genesis uses tactics as a search strategy to *eliminate* constraints

No Edge Tactic ensures no intermediate edge switch



Tactics: Algorithmic Properties

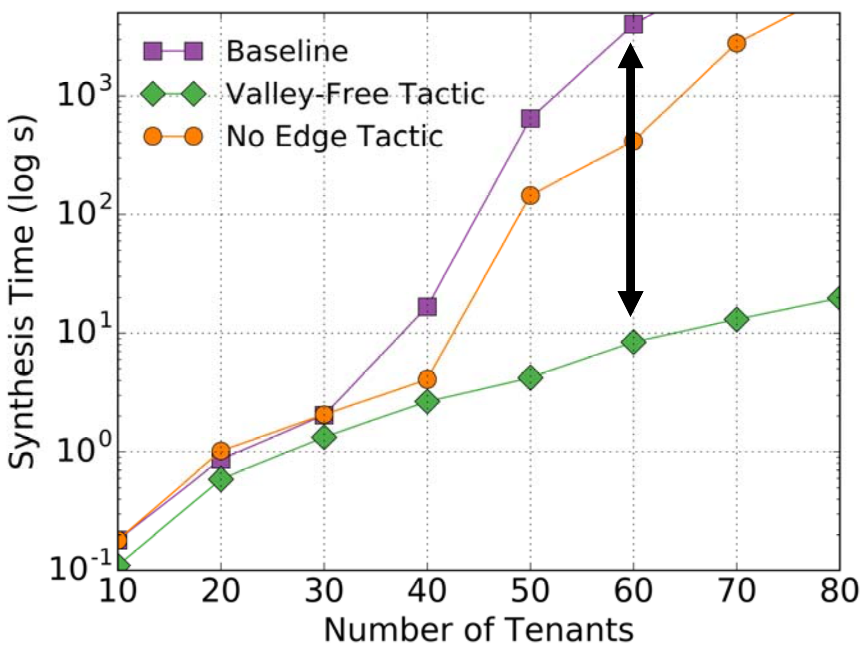
- Specified using a restricted subset of regular expressions
- Sound and Complete algorithm for enforcing them
- Policy-agnostic
- The operator can develop a repository of tactics based on their topology

Tactics: Evaluation

Multi-tenant isolation workload

Valley-Free Tactic and No Edge Tactic

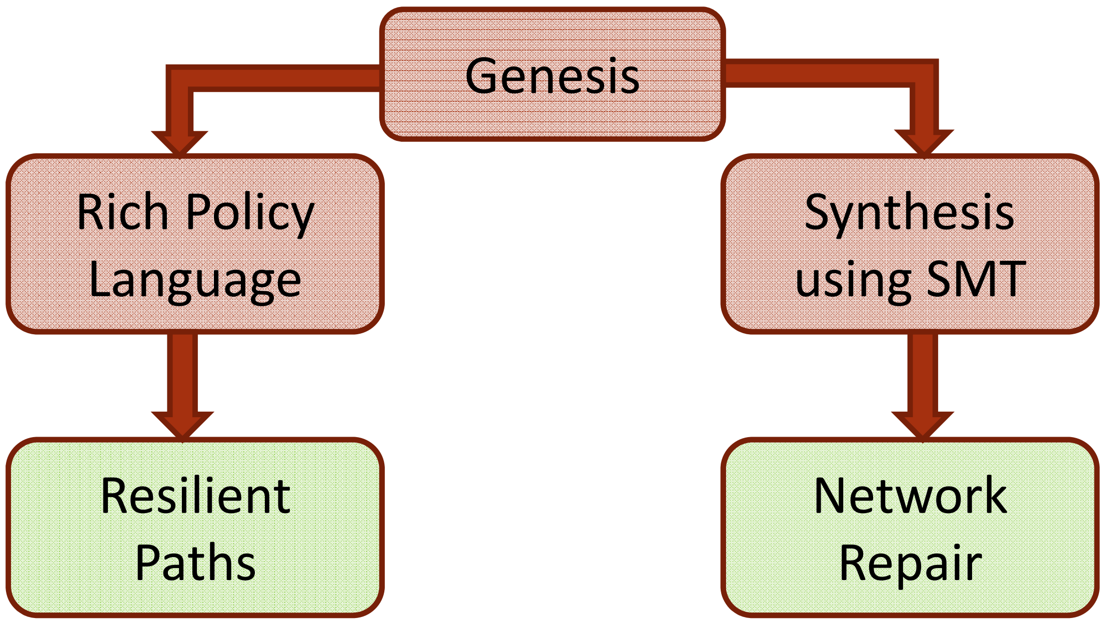
Valley-Free Tactic speedup: 400x



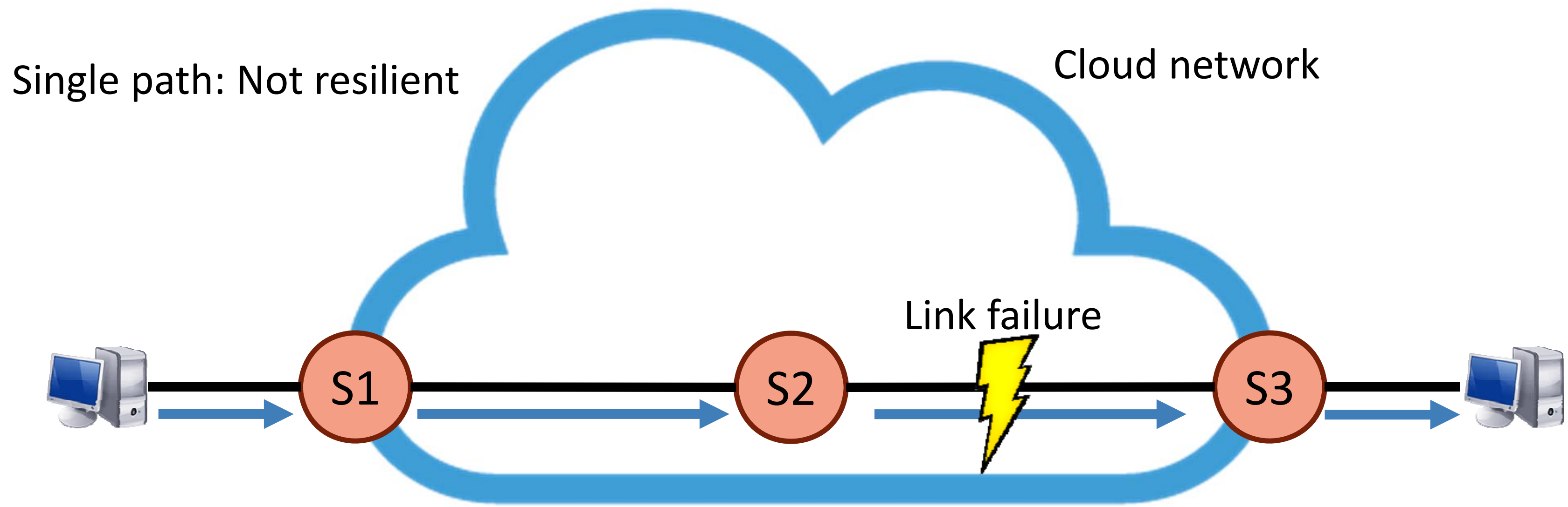
Outline of the Talk

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

Genesis Extensions



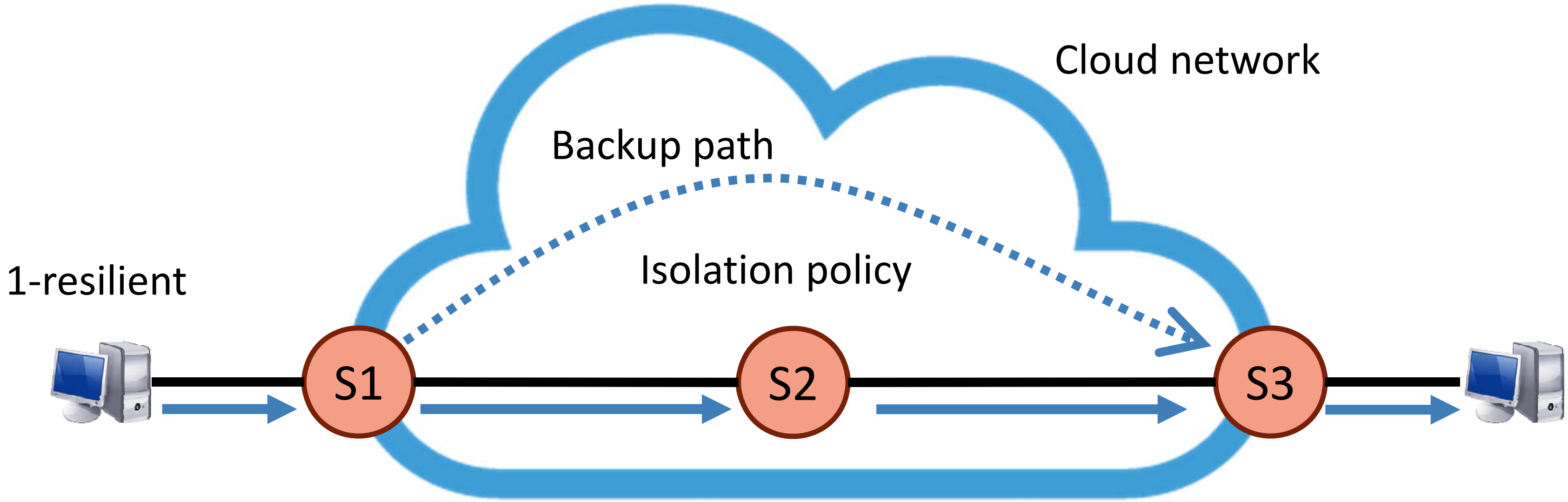
Network Resilience



t-resilience: For events under **t arbitrary link** failures,

there exists a valid path

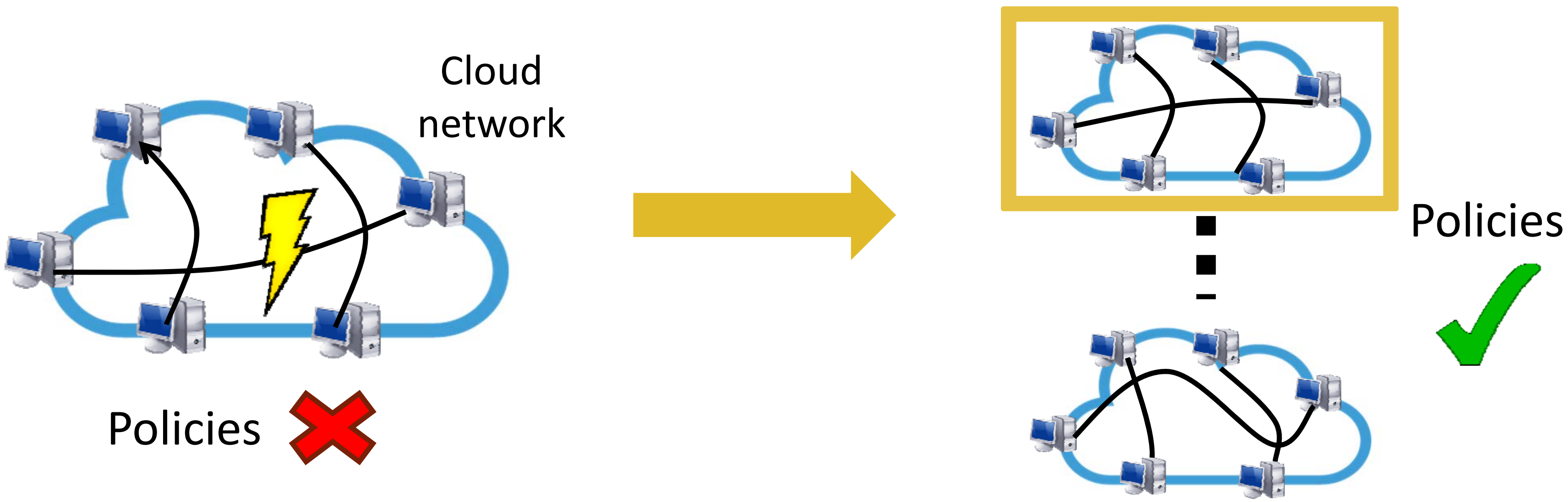
Policy-compliant Resiliency



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

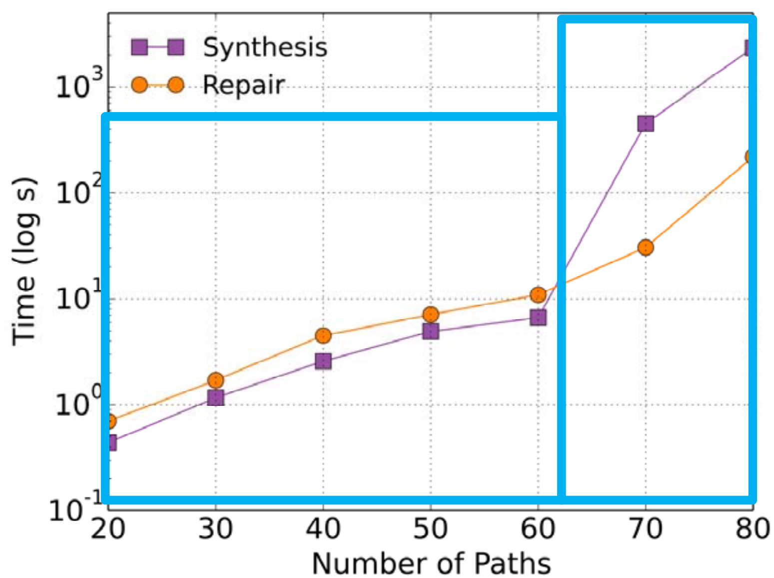
Genesis uses MaxSMT

Network Repair Evaluation

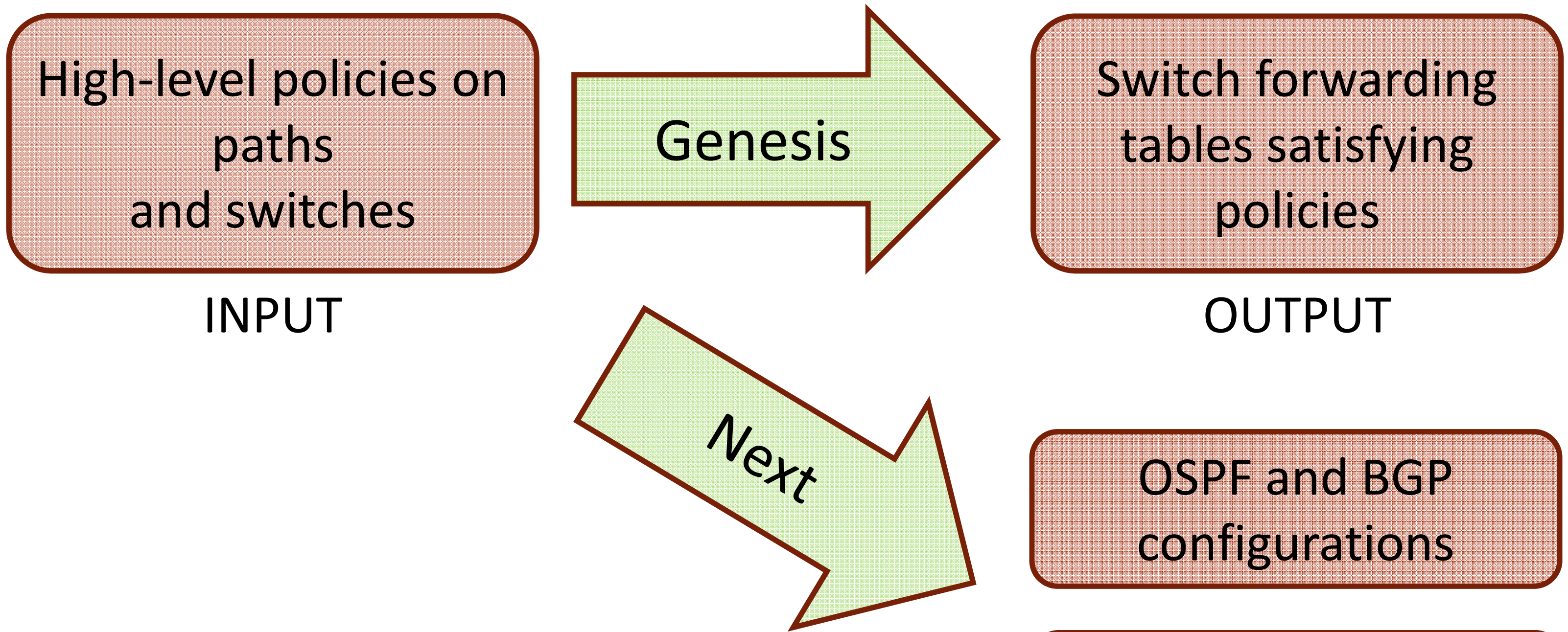
Multi-tenant isolation workload

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
INPUT
Genesis
Switch forwarding tables satisfying policies
OUTPUT
Next
OSPF and BGP configurations
Efficient optimal repair