ACORN: Network Control Plane Abstraction using Route Nondeterminism

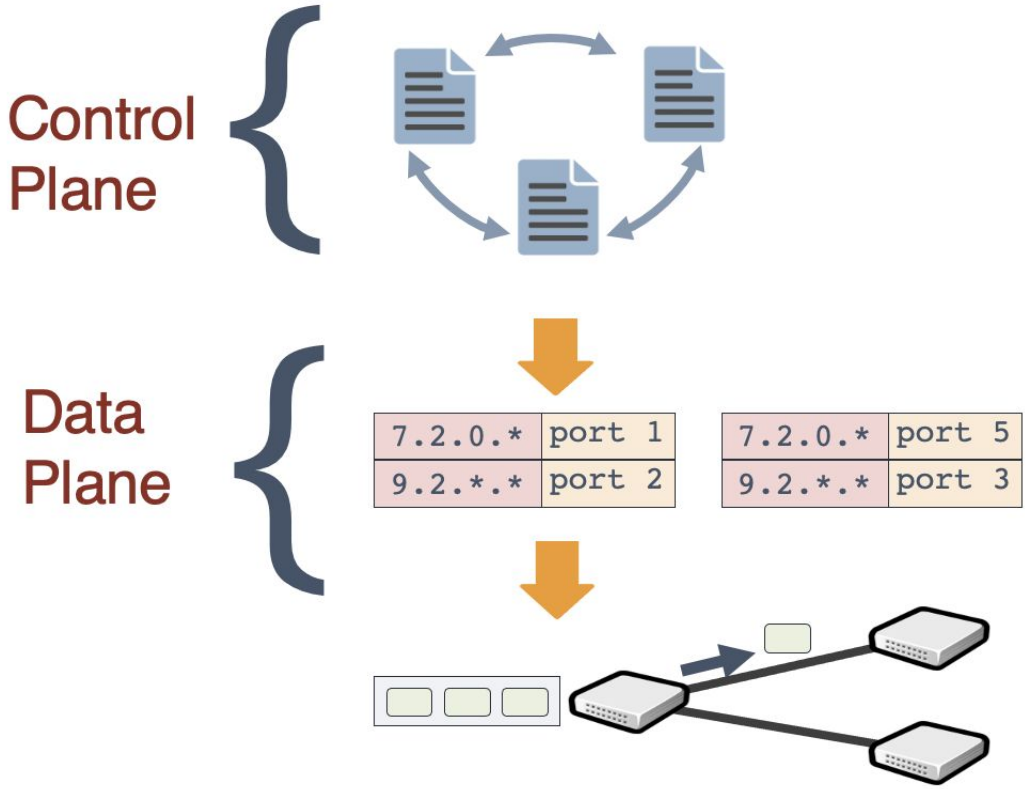
Divya Raghunathan
Princeton University

Ryan Beckett
Microsoft Research

Aarti Gupta
Princeton University

David Walker
Princeton University

Network design



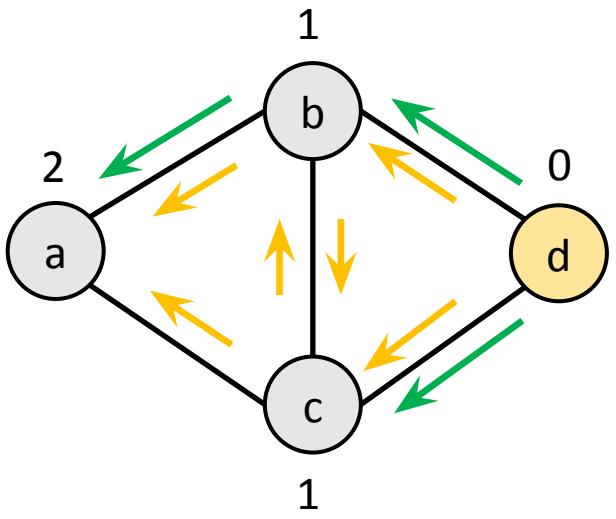
Protocol + Configuration

Routing protocols RIP, OSPF, BGP, …

Forwarding Tables

Runtime Behavior

Network control plane



- The origin sends an initial route announcement
- On receiving a route announcement:
  - a router processes it as per the configurations
  - selects the best route announcement received
  - forwards the selected one to its neighbors, after processing it
- At convergence, the network will be in a *stable state*

Network misconfigurations are common

How a coding error caused Rogers outage that left millions without service

ALEXANDRA POSADZKI >
PUBLISHED JULY 25, 2022

Major Fastly outage brings down much of the Internet: Amazon, Twitch, Reddit, and more offline

The Verge, The Guardian, and others go offline

June 08, 2021 By: Sebastian Moss Comment

Microsoft: Misconfigured Network Device Caused Azure Outage

A misconfigured network device caused Thursday's outage for the Windows Azure cloud computing platform, Microsoft said Friday. The downtime left the Azure Compute service unavailable to cstomers in some parts of Europe for more than two hours.

A Cloudflare outage broke large swathes of the internet

Jun 21, 2022

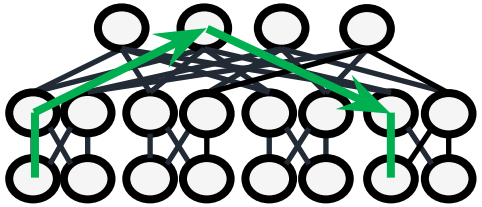
BGP super-blunder: How Verizon today sparked a 'cascading catastrophic failure' that knackered Cloudflare, Amazon, etc

'Normally you'd filter it out if some small provider said they own the internet'

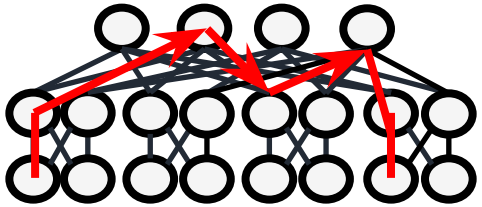
Kieren McCarthy in San Francisco Mon 24 Jun 2019 // 19:01 UTC

Network verification

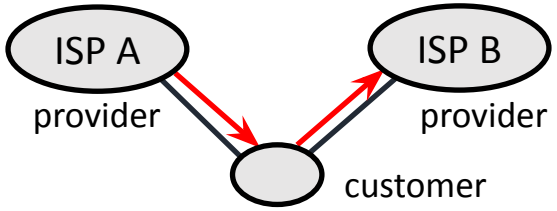
Properties



Reachability

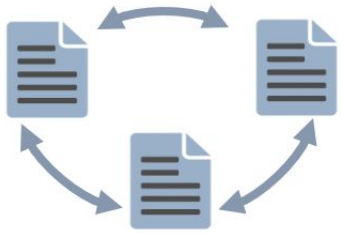


Valley-free



No-transit

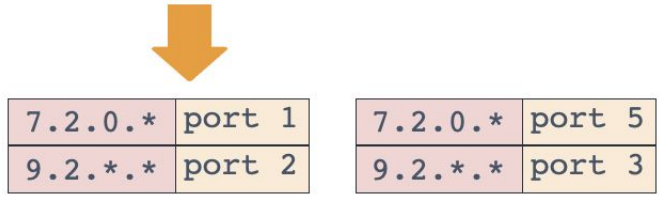
Control Plane



Control plane verifiers

- analyze router configurations that determine the forwarding rules
- have limited scalability, especially SMT-based approaches

Data Plane



Data plane verifiers

- analyze a snapshot of the network given forwarding rules
- scale to large networks (≈10,000 routers)

Goal: Improve scalability

Solve an *abstract* verification problem

Abstract certain network features while preserving soundness

SMT encoding

Symbolic graph-based encoding

Use SMT solvers with specialized theories, e.g. MonoSAT [Bayless et al. AAAI 2015]

Abstract the route selection procedure

- Best route is not always needed to verify properties (e.g., reachability)
  - Selected route should comply with network policy

  => Continue to model route filters *precisely*

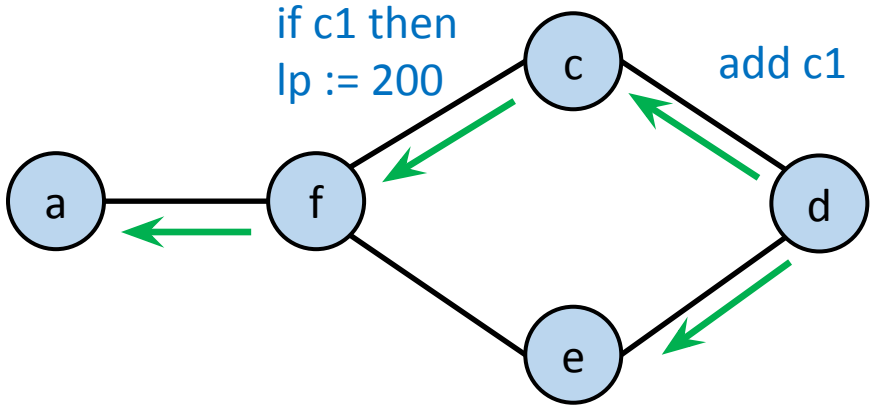
- Modeling route selection is expensive
  - Especially for complex policy-based routing protocols like BGP
  - One main difference between data plane and control plane

  => Get closer to performance of data plane verifiers by abstracting route selection

Example: Verifying reachability

Network N

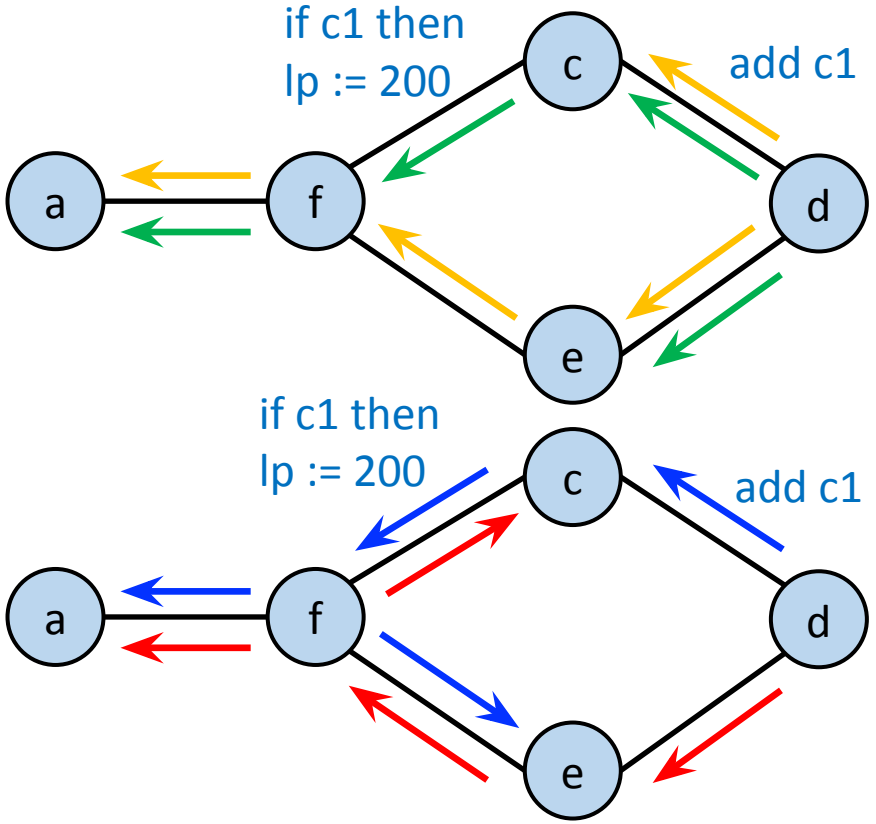
Each router selects the best route



a has a route to **d**

Abstract network N’

Each router selects any route

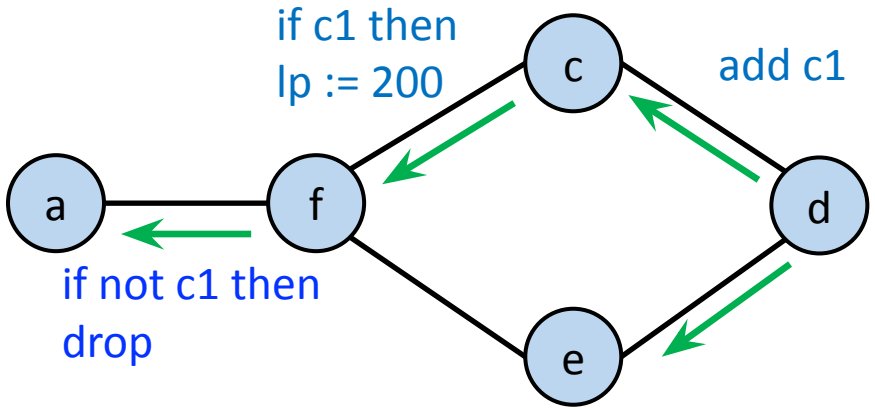


a has a route to **d** ✓

Example: Verifying reachability

Network N

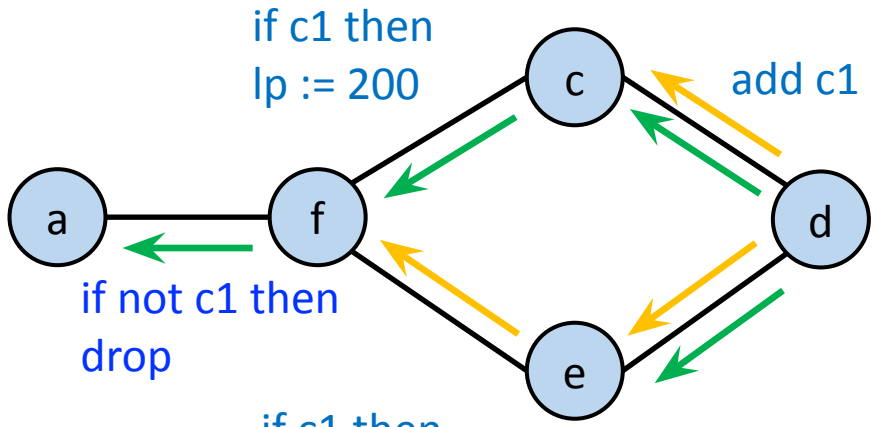
Each router selects the best route

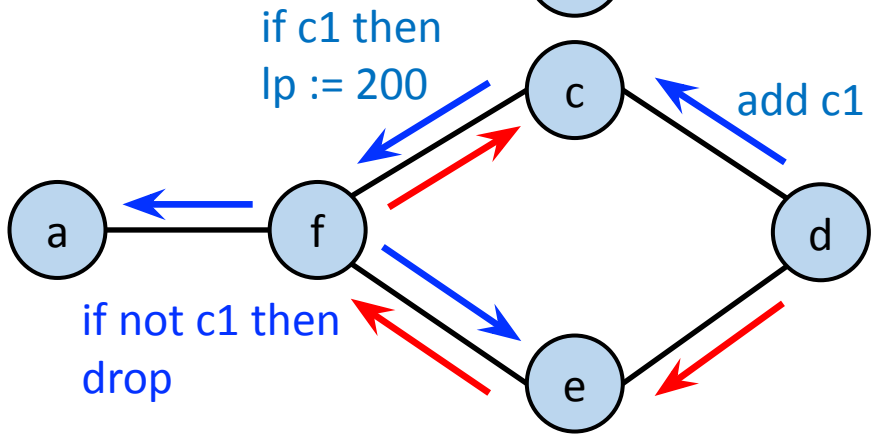


a has a route to **d**

Abstract network N'

Each router selects any route





a doesn't have a route to **d** ✗ **(false positive)**

For more precision,
we define a *hierarchy* of abstractions

Background: Stable Routing Problem (SRP)

$(G, A, a_d, \prec, trans)$

$G = (V, E, d)$
Network topology

Protocol →
- A: set of attributes
  Route announcements, ∞: no route, $A_\infty = A \cup \{\infty\}$
- $\prec \subseteq A_\infty \times A_\infty$
  Route selection: $a_1 \prec a_2 \Rightarrow a_1$ is preferred

Protocol + configuration →
- $a_d \in A$
  Initial route
- $trans: E \times A_\infty \to A_\infty$
  Route modification / dropping

SRP solutions $\mathcal{L}: V \to A_\infty$ represent stable states at convergence

[Beckett et al. SIGCOMM 2018]

Formalizing NRC abstractions via abstract SRP

$$(G, A, a_d, \prec', trans)$$

$G = (V, E, d)$
Network topology

Protocol →
- A: set of attributes
  Route announcements, ∞: no route, $A_\infty = A \cup \{\infty\}$

Protocol + abstraction →
- $\prec' \subseteq A_\infty \times A_\infty$
  Abstract route selection: $a_1 \prec' a_2 \Rightarrow a_1$ is preferred

Protocol + configuration →
- $a_d \in A$
  Initial route
- $trans: E \times A_\infty \to A_\infty$
  Route modification / dropping

SRP solutions $\mathcal{L}: V \to A_\infty$ represent stable states at convergence

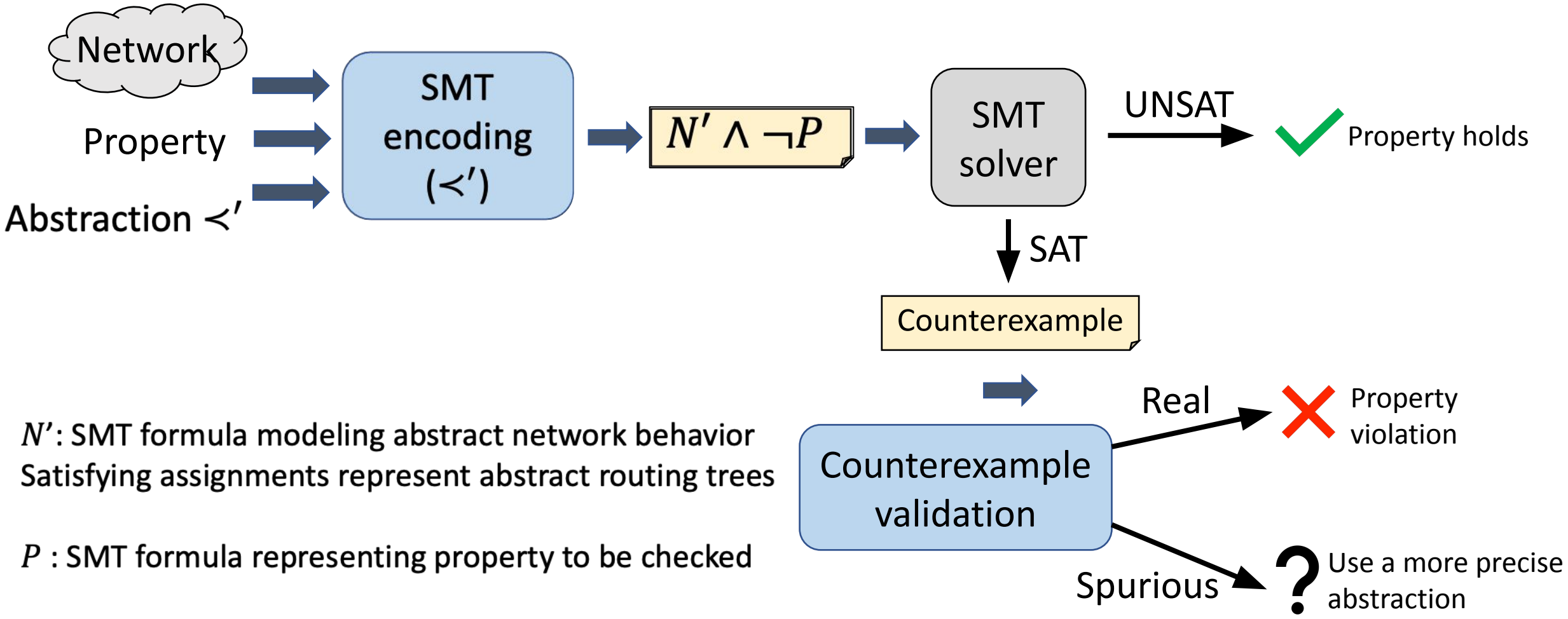
Soundness of NRC abstractions

Abstract SRP S' over-approximates corresponding SRP S

Theorem: Any solution of S, $\mathcal{L}: V \to A_\infty$ is a solution of S'

Proof uses the requirement that the minimal route according to $\prec$ is also minimal according to $\prec'$

SMT-based verification with NRC abstraction



N': SMT formula modeling abstract network behavior
Satisfying assignments represent abstract routing trees

P : SMT formula representing property to be checked

Goal: Improve scalability

Solve an *abstract* verification problem

Abstract certain network features while preserving soundness

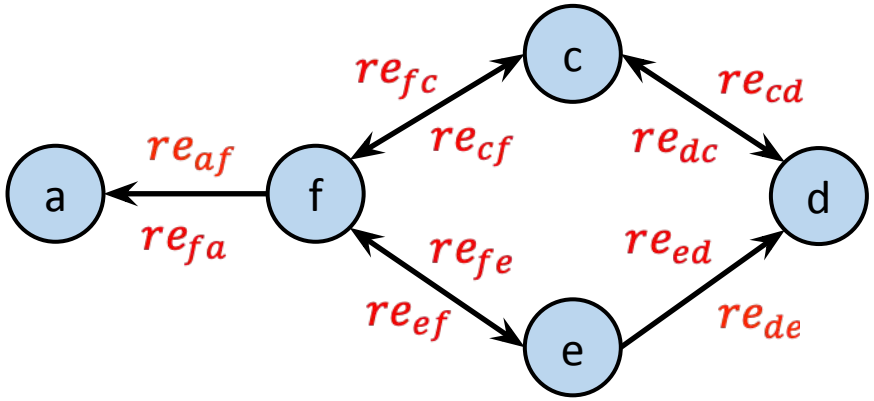
SMT encoding

Symbolic graph-based encoding

Use SMT solvers with specialized theories, e.g. MonoSAT [Bayless et al. AAAI 2015]

Background: MonoSAT SMT solver

- SMT solver with support for graph-based reasoning
- Uses symbolic graphs, graphs with a Boolean variable per edge



Symbolic graph $G_{RE} = (G, RE)$

$G = (V, E)$

$RE = \{re_{uv} \mid (u, v) \in E\}$

Reachability predicate: $G_{RE}.reaches(u, v)$

Formula F over RE and other variables, with graph predicates like reachability

Satisfying assignment corresponds to a subgraph of G (symbolic graph solution)

[Bayless et al. AAAI 2015]

Symbolic graph-based encoding

Encode abstract SRP S' with SMT formula N'

Symbolic graph solutions = solutions of abstract SRP S'

Four types of constraints:

1. Routing choice constraints
2. Route availability constraints
3. Attribute transfer and route filtering constraints
4. Solver-specific constraints for route availability

   Reachability predicate used in MonoSAT

Symbolic graph-based encoding

Routing choice constraints

Each node u chooses a neighbor v or None

$$\left(\bigvee_{(v,u)\in E} nChoice_u = nID(u,v) \right) \vee nChoice_u = None_u$$

$$nChoice_u = nID(u,v) \leftrightarrow re_{vu}$$

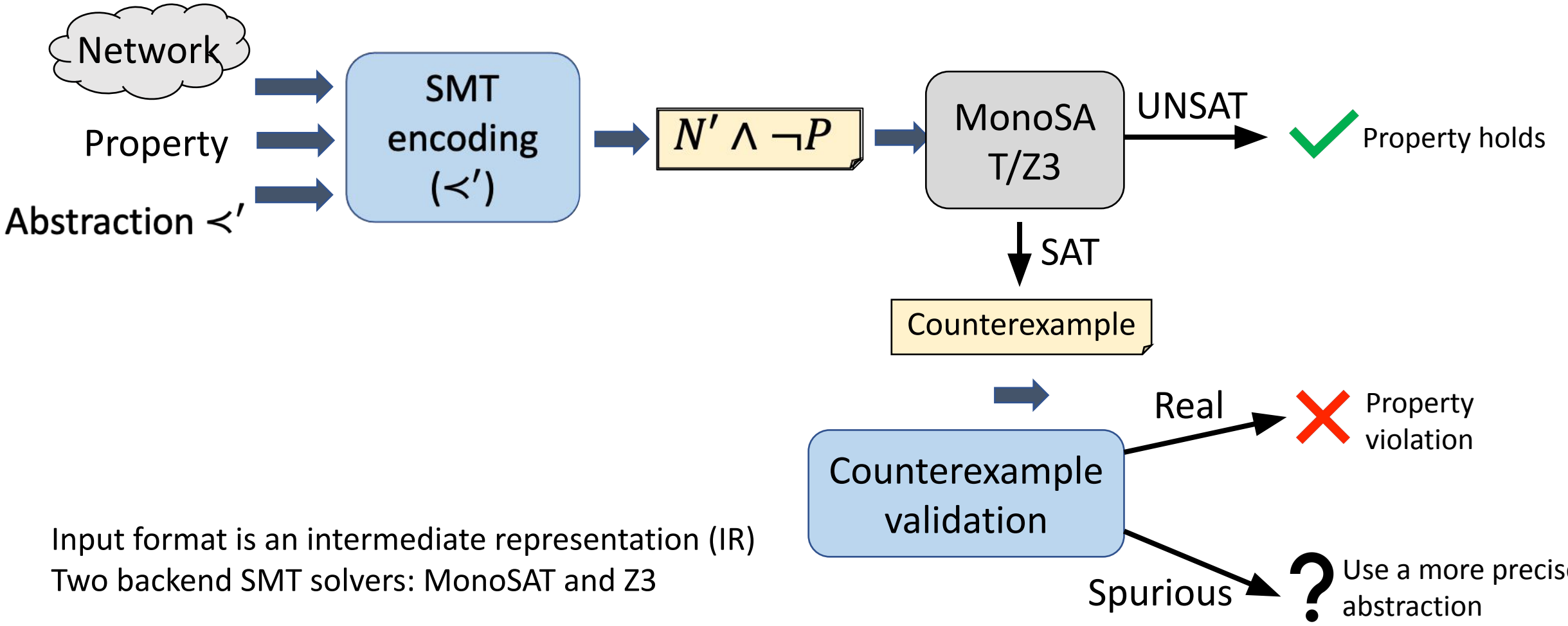
Symbolic edge variable

Benefits of NRC and graph-based encoding

- Fewer variables
  - Route announcement fields used only in route selection are discarded
  - Community attribute (in BGP) sufficient for most policies evaluated

- Expensive transfers can become irrelevant during solver search
  - Once a symbolic edge variable is assigned true, other neighbors become irrelevant
  - Without abstraction, computing the best route requires transfers from all neighbors

ACORN prototype implementation



Input format is an intermediate representation (IR)
Two backend SMT solvers: MonoSAT and Z3

Evaluation

1. Relative performance of NRC abstractions (with / without)
2. Relative performance of graph theory capable SMT solver (MonoSAT / Z3)

Four experiment settings:

- **abs_mono**: NRC abstraction using MonoSAT
- **abs_z3**: NRC abstraction using Z3
- **mono**: no abstraction using MonoSAT
- **z3**: no abstraction using Z3

Benchmarks

1. Data center examples with FatTree topology (to evaluate scalability)
   - Synthetic benchmarks with shortest-path and valley-free policies

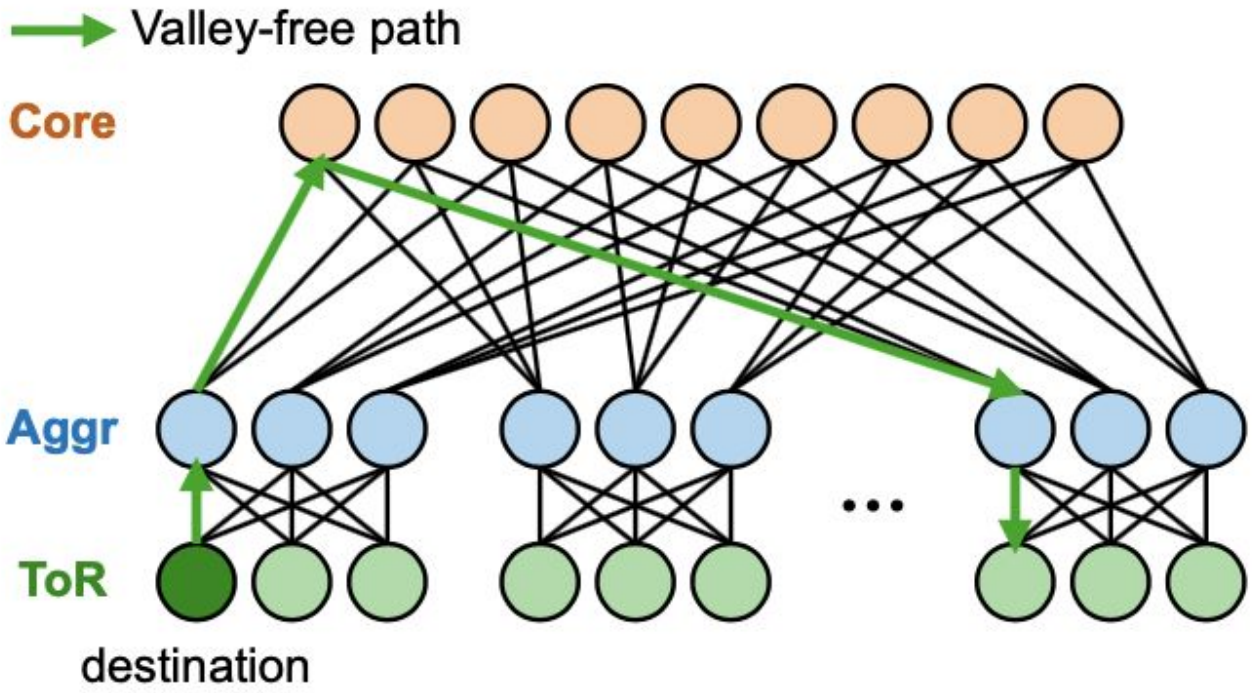
2. Wide Area Network (WAN) examples with real-world topologies
   - Topology zoo examples, which we annotated with business relationships
   - BGPStream examples, annotated using the CAIDA AS relationships dataset

All our benchmarks are publicly available:

https://github.com/divya-urs/ACORN_benchmarks

Machine details: 2.3 GHz Intel i7 processor, 16 GB RAM

FatTree network with valley-free policy



c: community attribute (2 bits)

Aggr → Core

if c == 0 then c = 1
else drop route

Aggr → ToR

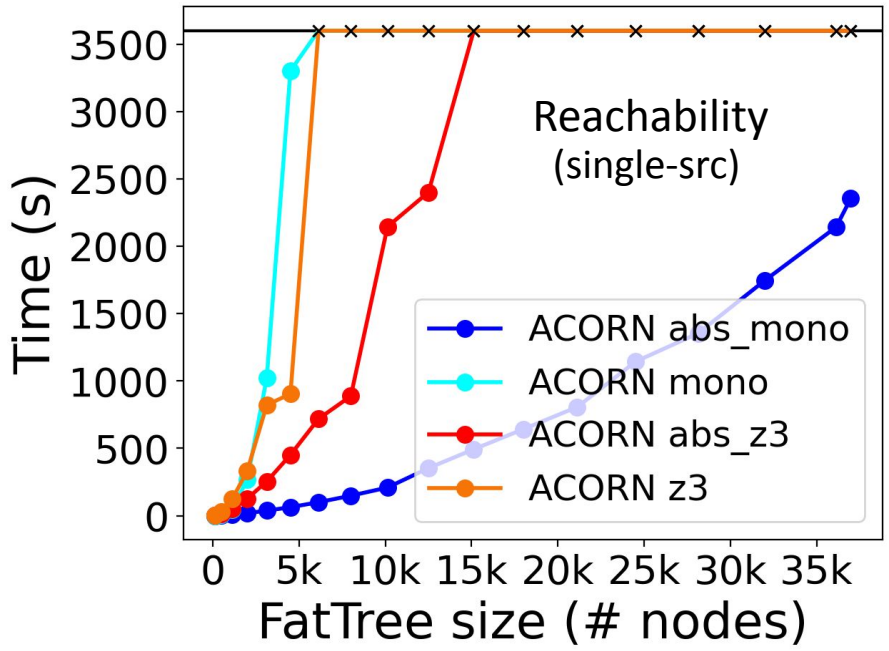
if c == 0 then c = 1
else if c == 1 then c = 2
else c = 3

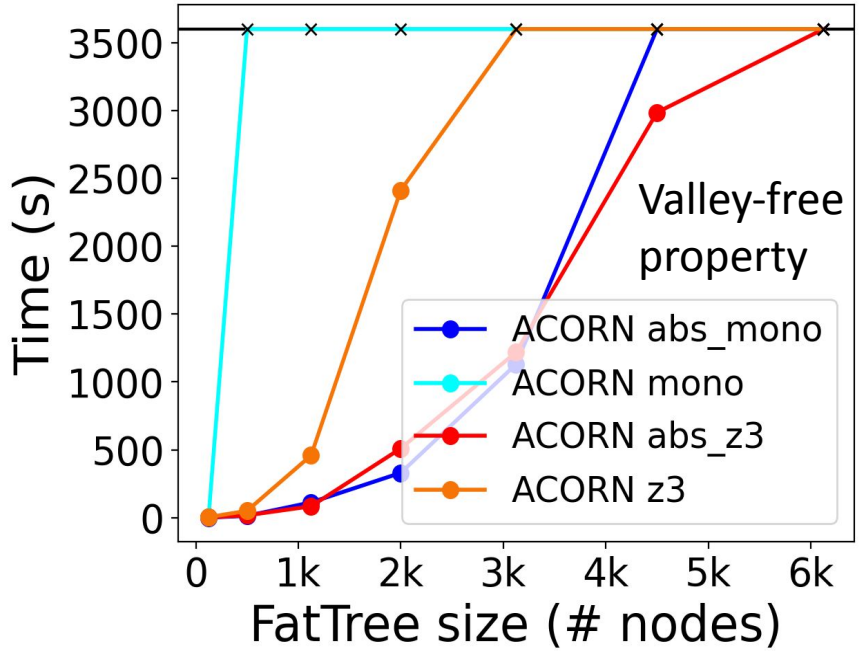
ToR → Aggr

if c != 0 then drop route

c = 0: route has 0 Aggr nodes
c = 1: route has 1 Aggr nodes
c = 2: route has 2 Aggr nodes
c = 3: route has ≥ 3 Aggr nodes

Results for FatTrees with valley-free policy



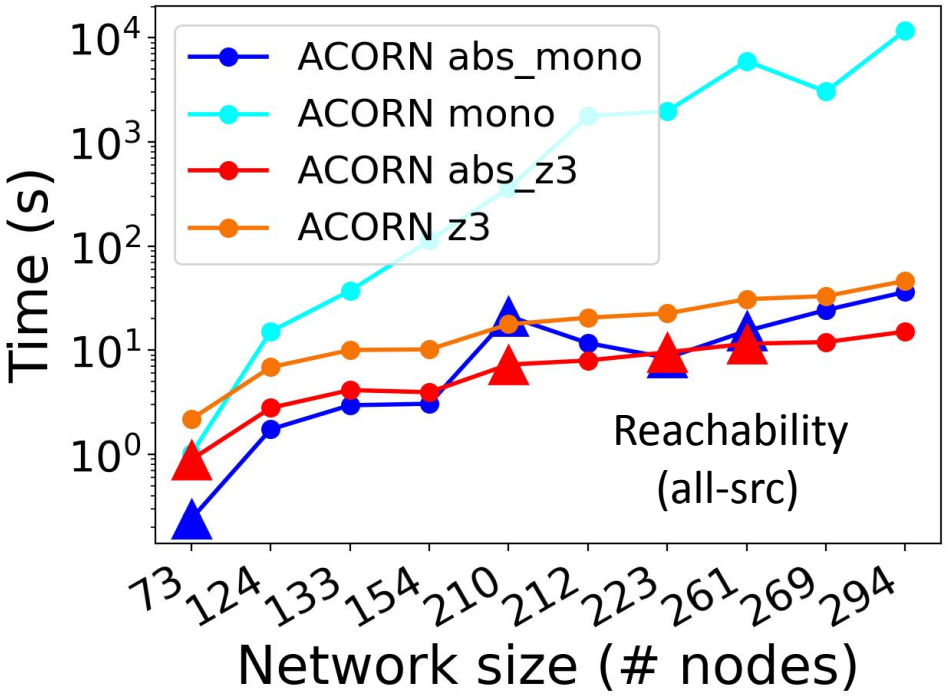


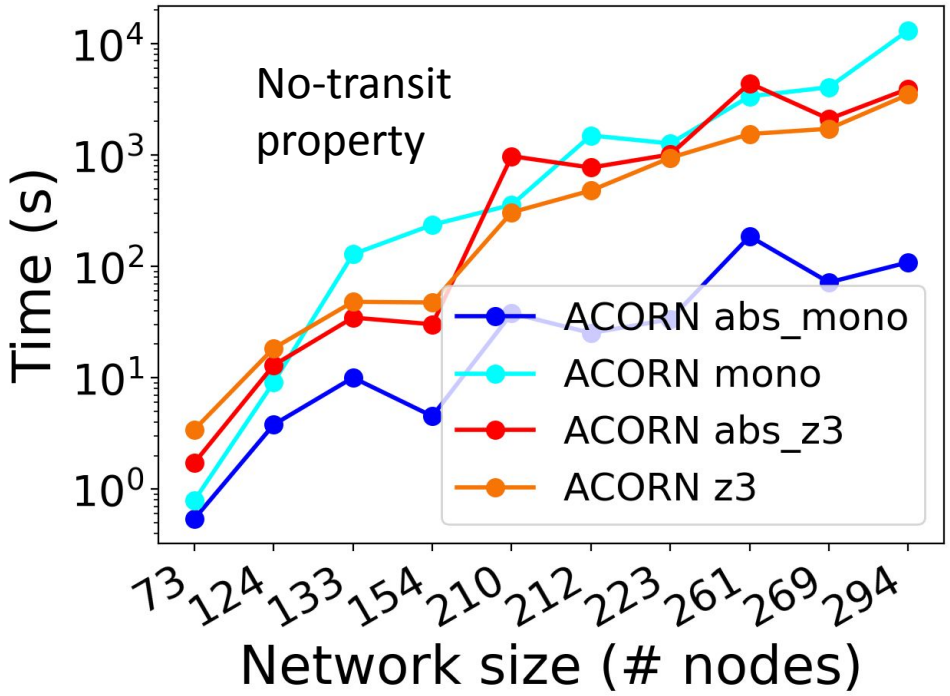
Abstract settings verify both properties without false positives

Abstraction improves performance in *all* networks (up to 52x speedup for MonoSAT)

abs_mono verifies reachability for a FatTree with 36,980 routers in 40 mins

Results for BGPStream examples



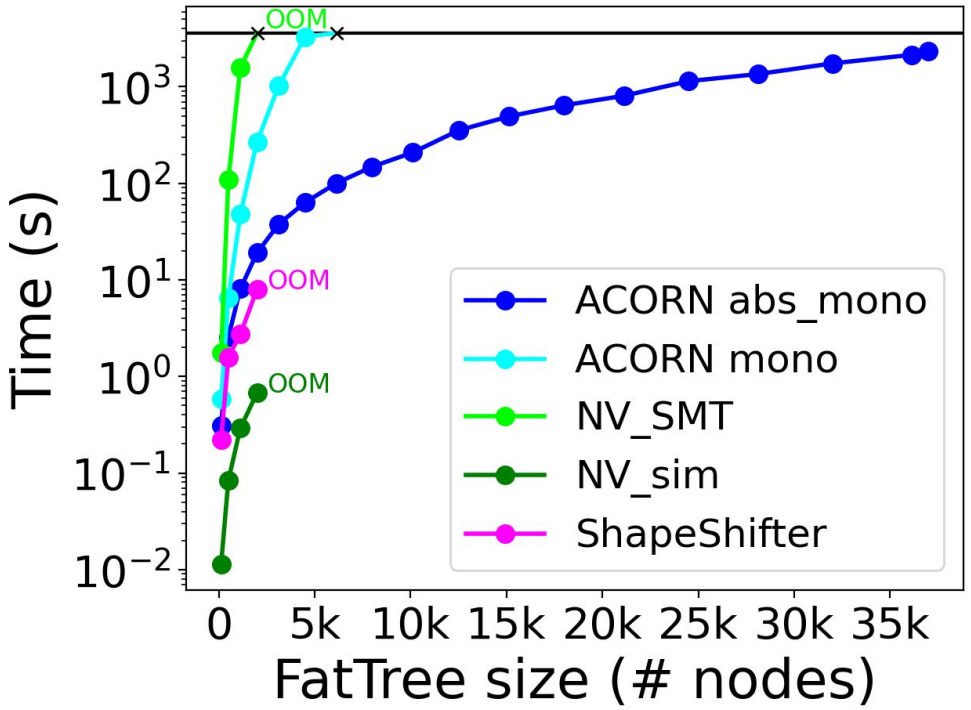


Abstract settings verify *no-transit property* in **all** networks

Abstract settings verify *reachability* in **6/10** networks

Remaining 4/10 are verified using a more precise abstraction

Comparison with other tools



Reachability (single-src) on FatTree benchmarks with valley-free policy

- NV uses MTBDD-based simulation and SMT
- ShapeShifter uses BDD-based simulation with abstract interpretation
- NV and ShapeShifter *run out of memory* for networks with **> 3000 nodes**
- ACORN scales to **≈37,000 nodes** for reachability

Summary

- Nondeterministic Routing Choice (NRC) abstraction hierarchy
  Formalized using the SRP model and proved sound for verification
- Symbolic graph-based SMT encoding
  Leverages SMT solvers with graph theory support as well as standard SMT solvers
- NRC abstractions can verify realistic policies and improve scalability
  Could verify reachability for ≈37,000 routers within an hour