Config2Spec:
Mining Network Specifications from Network Configurations

Rüdiger Birkner, Dana Drachsler-Cohen, Martin Vechev, Laurent Vanbever

nsg.ee.ethz.ch

NSDI’20
February, 27 2020

ETH zürich
Many tools are available that allow you to check that your network behaves as intended.

Standard recipe:

1. Upload configurations
2. Define specification
3. Run the tool
4. Iterate & deploy
Many tools are available that allow you to check that your network behaves as intended.

Standard recipe:

1. Upload configurations
2. Define specification
3. Run the tool
4. Iterate & deploy
Definition
The specification of a network is the set of all policies that hold...

Set of policies

reachability(r1,p1)
waypoint(r3,r1,p2)
reachability(r5,p2)
...
loadbalancing(r3,p2)

Topology
Definition
The specification of a network is the set of all policies that hold...

Set of policies

- reachability(r1, p1)
- waypoint(r3, r1, p2)
- reachability(r5, p2)

... loadbalancing(r3, p2)

Topology
Definition

The specification of a network is the set of all policies that hold...

Set of policies

- reachability(r1, p1)
- waypoint(r3, r1, p2)
- reachability(r3, p2)
- ...
- loadbalancing(r3, p2)

Topology

- r1
- r2
- r3
- r4
- r5
- p1
- p2
**Definition**

The specification of a **network** is the **set of all policies** that hold under a given **failure model**.

**Set of policies**

- reachability($r_1, p_1$)
- waypoint($r_3, r_1, p_2$)
- reachability($r_5, p_2$)
- loadbalancing($r_3, p_2$)
Definition

The specification of a **network** is the **set of all policies** that hold under a given **failure model**.

Set of concrete environments

Symbolic environment

Failure bound

\[ k = 2 \]
Definition

The specification of a network is the set of all policies that hold under a given failure model.

Set of policies

- reachability(r1, p1)
- waypoint(r3, r1, p2)
- reachability(r5, p2)
- ...
- loadbalancing(r3, p2)

Symbolic environment

Failure bound

\[ k = 2 \]
Writing the network’s precise specification is hard

Dr Heidy Khlaaf
@HeidyKhlaaf

In the past three years of working on large safety critical systems, I’ve learned that verification isn’t the real problem, but it’s writing specifications. Don’t @ me.
twitter.com/Conaw/status/1…

Putting network verification to good use

Ryan Beckett
Microsoft Research

Ratul Mahajan
University of Washington
Intentionet

… However, outside of a handful of large cloud computing providers, the use of network verification is still sparse.

Standard recipe:

1. Upload configurations
2. Define specification
3. Run the tool
4. Iterate & deploy
Internet2’s specification with its 10 routers consists of ~4000 policy predicates.
Internet2’s specification with its 10 routers consists of ~4000 policy predicates.

Imagine writing that specification by hand.
Introducing

Config2Spec
Config2Spec automatically mines the network’s full specification from its configuration and the given failure model

Input

Network Configuration

Failure Model

Output

Network Specification

loadbalancing(4, p2)
reachability(1, p1)
reachability(1, p2)
...
reachability(4, p2)
reachability(5, p2)
Config2Spec:
Mining Network Specifications from Network Configurations

1. Baseline approaches
   one search space at a time

2. Our approach
   the best of both worlds

3. Evaluation
   scales to realistic networks
Config2Spec: Mining Network Specifications from Network Configurations

1. **Baseline approaches**
   one search space at a time

2. **Our approach**
   the best of both worlds

3. **Evaluation**
   scales to realistic networks
Mining a network specification involves exploring two exponential search spaces

all concrete environments

reachability(r1,p1)
waypoint(r3,r1,p2)
...  
loadbalancing(r5,p2)

all possible policies
Mining a network specification involves exploring two exponential search spaces

data plane analysis  ×  control plane verification
data plane analysis × control plane verification
Data plane analysis tools allow to find all the policies that hold for a single concrete environment.
The network specification is the intersection of the policies that hold for every concrete environment.
The network specification is the intersection of the policies that hold for every concrete environment.
data plane analysis × control plane verification
Control plane verification tools determine whether a policy holds for the entire failure model.

$k = 2$

reachability($r_1, p_1$)

Single policy

Control plane verification

Verification result
The network specification is the set of policies that the verifier determined to hold for the failure model.
The network specification is the set of policies that the verifier determined to hold for the failure model.

Network specification under -approximation of the specification.
Both techniques have pros and cons

<table>
<thead>
<tr>
<th>Approach</th>
<th>Data Plane Analysis</th>
<th>Control Plane Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all policies for</td>
<td>one policy for the</td>
</tr>
<tr>
<td></td>
<td>one concrete env.</td>
<td>entire failure model</td>
</tr>
</tbody>
</table>

| Good at pruning | Dense Violations    | Sparse Violations           |
Violations are policies that are not part of the specification

Policies that hold for almost no envs.

dense violation

waypoint(r5, r2, p2)

sparse violation

reachability(r1, p2)

k = 3
Both techniques have pros and cons

<table>
<thead>
<tr>
<th>approach</th>
<th>data plane analysis</th>
<th>control plane verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>all policies for</td>
<td>good at pruning</td>
<td></td>
</tr>
<tr>
<td>one concrete env.</td>
<td>dense violations</td>
<td>sparse violations</td>
</tr>
<tr>
<td>one policy for the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>entire failure model</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What about combining them?
Config2Spec:
Mining Network Specifications from Network Configurations

1. Baseline approaches
   one search space at a time

2. Our approach
   the best of both worlds

3. Evaluation
   scales to realistic networks
Config2Spec mines the network’s full specification from its configuration and the required failure tolerance.

- **Input**
  - Network Configuration
  - Failure Model

- **Config2Spec**
  - Data Plane Analysis
  - Control Plane Verification
  - Predictor
  - Infer a specification
  - Verify that specification

- **Output**
  - Network Specification
    - loadbalancing(4, p2)
    - reachability(1, p1)
    - reachability(1, p2)
    - ...
    - reachability(4, p2)
    - reachability(5, p2)
Thanks to combining the two approaches, \textit{Config2Spec} is precise

Data plane analysis
- over-approximation

Control plane verification
- under-approximation
Step-by-step from all existing policies to the network’s specification
By performing data plane analysis on a topology, Config2Spec refines the space of candidate policies.
By performing data plane analysis on a topology, Config2Spec refines the space of candidate policies.
By performing data plane analysis on a topology, Config2Spec refines the space of candidate policies.
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.

Policy #1
reachability(1,p1)
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.

Policy #1
reachability(1,p1)
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.
With control plane verification, Config2Spec checks whether a candidate policy belongs to the specification.
When Config2Spec terminates, it is left with the specification.
Config2Spec can be improved further by two domain-specific techniques:

- policy-aware selection
- grouping and trimming
policy-aware selection

grouping and trimming
Data plane analysis has to reduce the candidate set to a minimum as fast as possible

**basic**
randomly pick concrete environments

**policy-aware**
pick concrete envs. based on the candidate set

Candidate set
reachability(r1, p2)
reachability(r2, p2)

Failure model
k=2

env #n
forwarding state
env #n+1
policy-aware selection

grouping and trimming
To be fast, control plane verification should be used as little as possible

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>verify each policy separately</td>
</tr>
<tr>
<td>trimming</td>
<td>leverage topology and failure model</td>
</tr>
<tr>
<td></td>
<td>can connectivity requirements be met?</td>
</tr>
<tr>
<td>grouping</td>
<td>verify similar policies at once</td>
</tr>
<tr>
<td></td>
<td>grouping is based on the destination</td>
</tr>
</tbody>
</table>
Config2Spec:
Mining Network Specifications from Network Configurations

1. Baseline approaches
   one search space at a time

2. Our approach
   the best of both worlds

3. Evaluation
   scales to realistic networks
<table>
<thead>
<tr>
<th>Question #1</th>
<th>How does Config2Spec scale to large-scale configurations?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question #2</td>
<td>How do the domain-specific techniques contribute to Config2Spec?</td>
</tr>
</tbody>
</table>
We fully implement Config2Spec and show its practicality

**Implementation**
5k lines of Python and Java
relying on Batfish [NSDI’15] and Minesweeper [Sigcomm’17]

**Methodology**
generated configs using NetComplete [NSDI’18]
employing OSPF, BGP

for a small, medium, and large network
with 33, 70, and 158 routers
Config2Spec mines the specification for realistic networks in few hours

- Data plane analysis
- Control plane verification
- OSPF
- BGP

Run time [h]
Large topology

\[ k = 1, 2, 3 \]
For failure models with few concrete environments, data plane analysis on its own provides fastest progress.
For failure models with a high failure bound, policy trimming reduces the candidate space significantly.
Config2Spec mines the specification for realistic networks in few hours
Policy-aware environment selection leads to smaller candidate sets with fewer samples than random.
Policy trimming and grouping allows to significantly reduce the number of queries for control plane verification.

- After data plane analysis: 100%
- After policy trimming: 16%
- After policy grouping: 3.5%

Medium topology, $k = 2$
Config2Spec:
Mining Network Specifications from Network Configurations

1 Baseline approaches
one search space at a time

2 Our approach
the best of both worlds

3 Evaluation
scales to realistic networks
Config2Spec:
Mining Network Specifications from Network Configurations

automatically learns a network's specification based on its configuration and failure model

scales thanks to the combination of the two approaches data plane analysis and control plane verification

the specification is useful beyond verification what-if analysis, config streamlining, network understanding
Config2Spec

Mining Network Specifications from Network Configurations

Rüdiger Birkner  Dana Drachslr-Cohen  Martin Vechev  Laurent Vanbever

nsg.ee.ethz.ch