Config2Spec:
Mining Network Specifications from Network Configurations

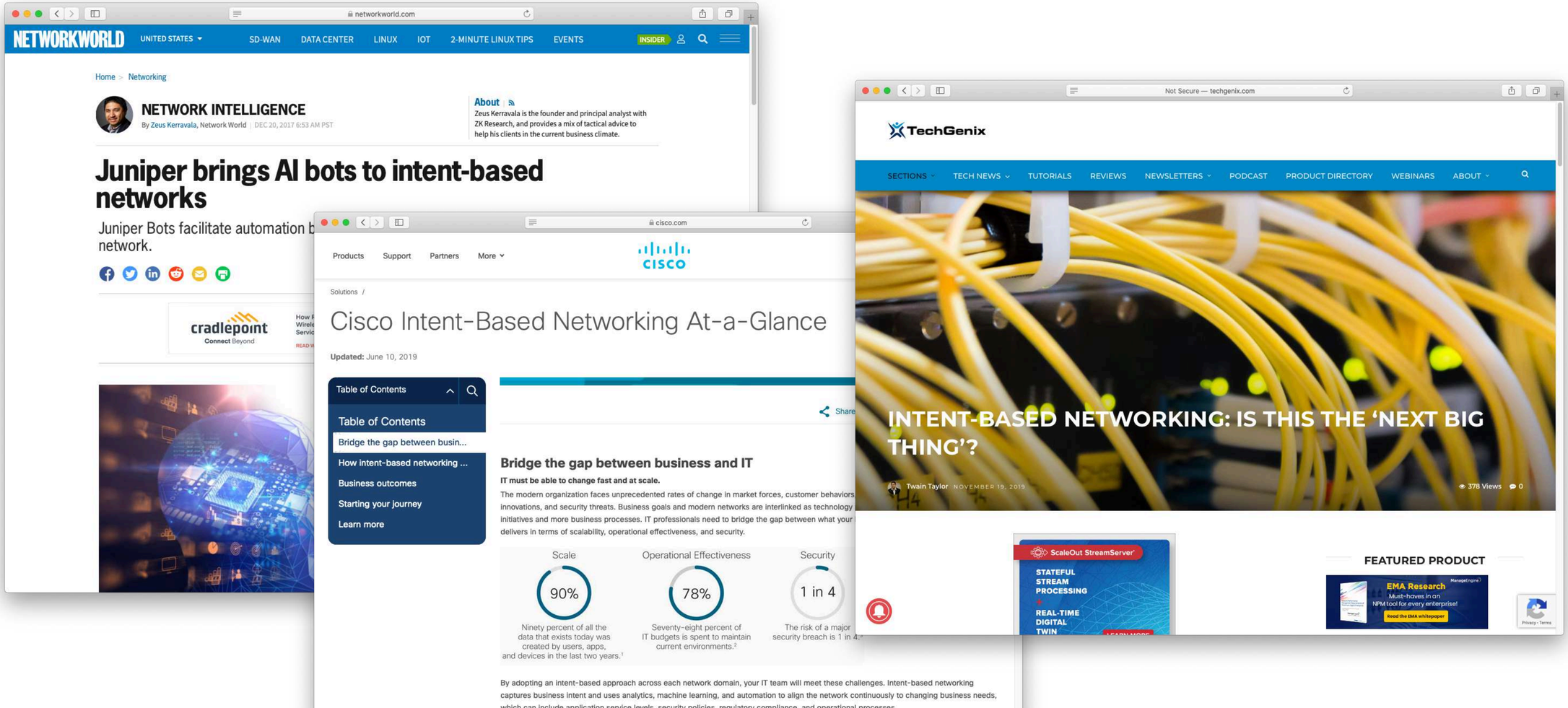
Rüdiger Birkner, Dana Drachsler-Cohen,
Martin Vechev, Laurent Vanbever

nsg.ee.ethz.ch

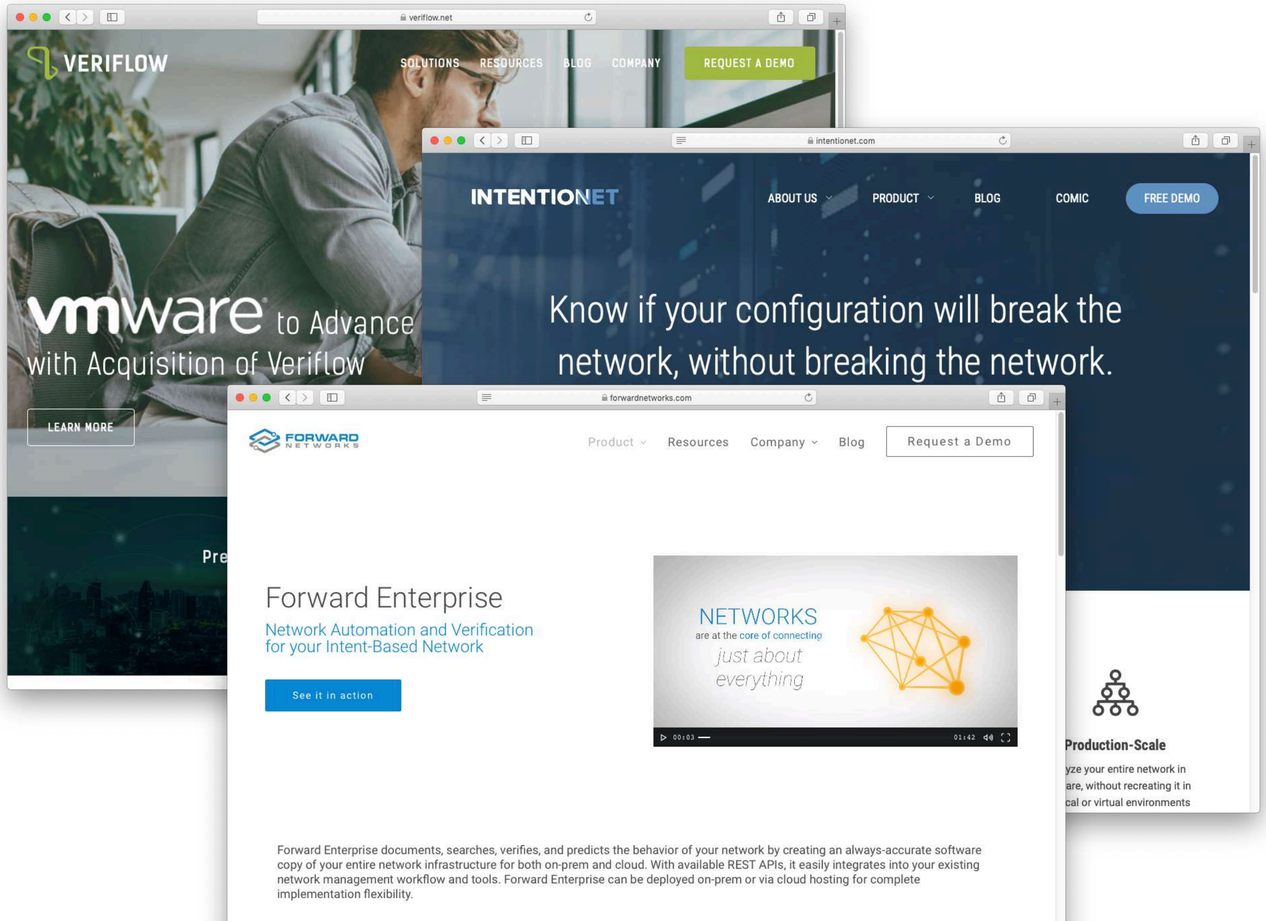
RIPE80
May, 11 2020

ETH zürich

Intent-based networking has been and still is one of the buzzwords in the community



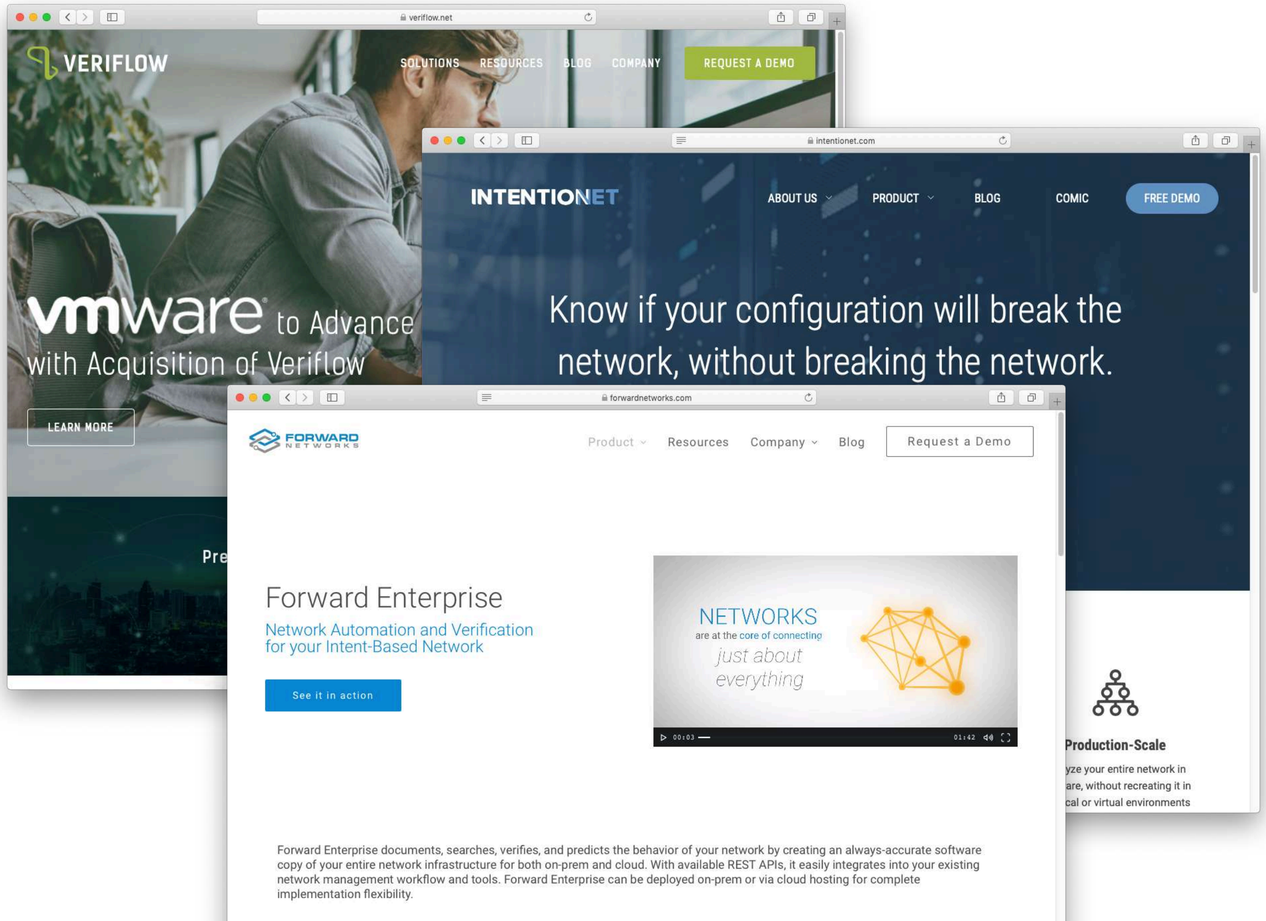
Many tools are available that allow you to check that your network behaves as intended



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 under a given **failure model**.

Definition

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

What needs to hold

Definition

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

What needs to hold

```
reachability(r1,p1)
waypoint(r3,r1,p2)
reachability(r5,p2)
            …
loadbalancing(r3,p2)
```

Definition

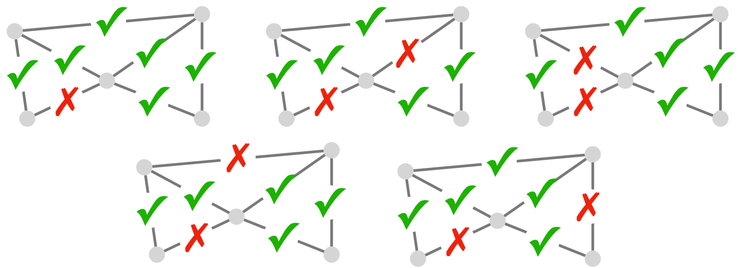
The specification of a **network** is
the **set of all policies** that hold under a given **failure model.**

when it needs to hold

Definition

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

when it needs to hold



Writing the network's precise specification is hard

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.

Config2Spec

Mining Network Specifications from Network Configurations

Rüdiger Birkner

Dana Drachsler-Cohen

Martin Vechev

Laurent Vanbever

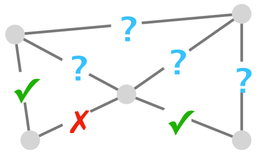
nsg.ee.ethz.ch

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

Input

Network Configuration

Failure Model



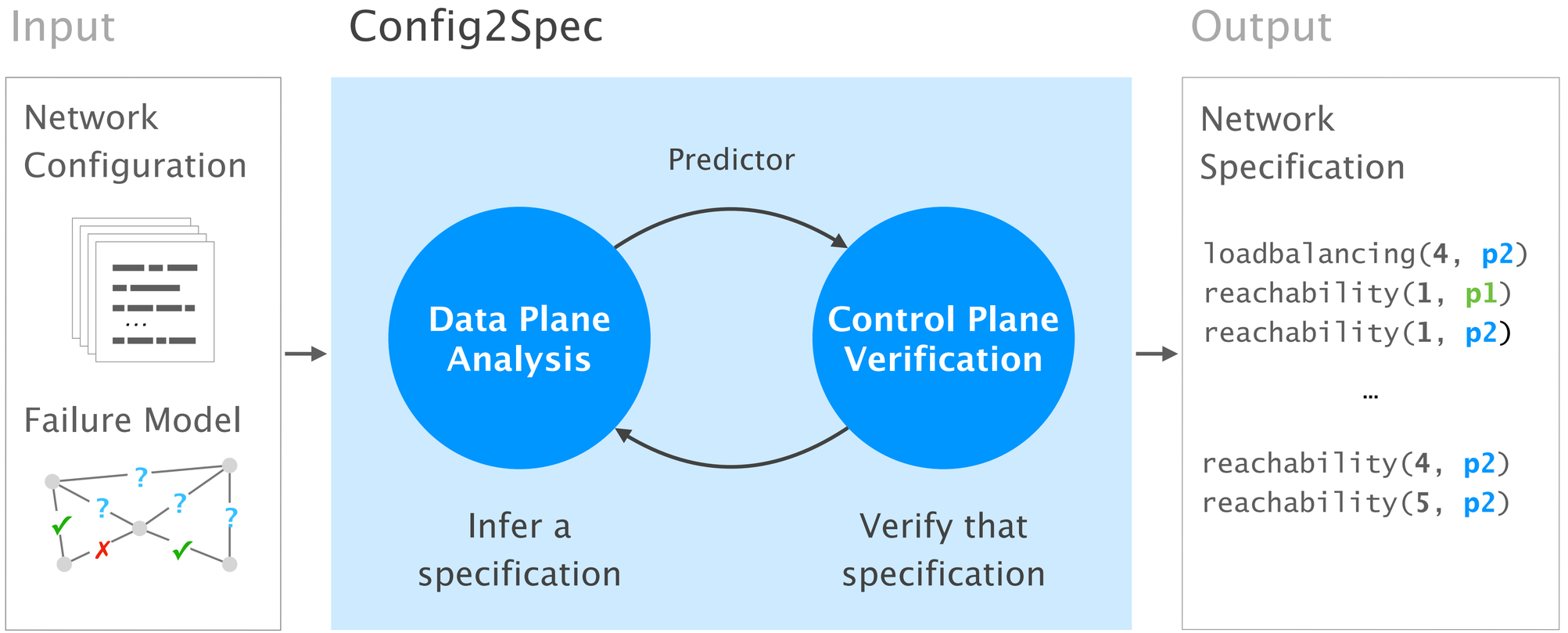
Config2Spec

Output

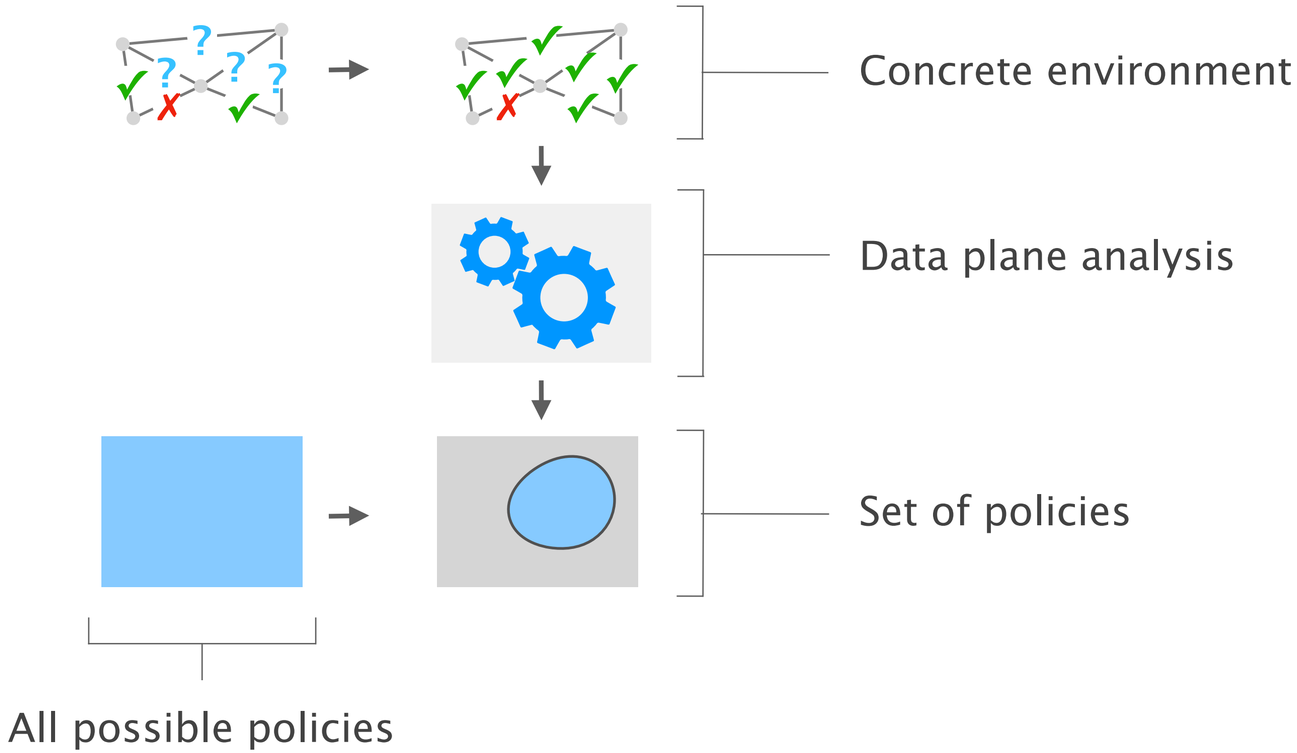
Network Specification

```
loadbalancing(4, p2)
reachability(1, p1)
reachability(1, p2)

...

reachability(4, p2)
reachability(5, p2)
```

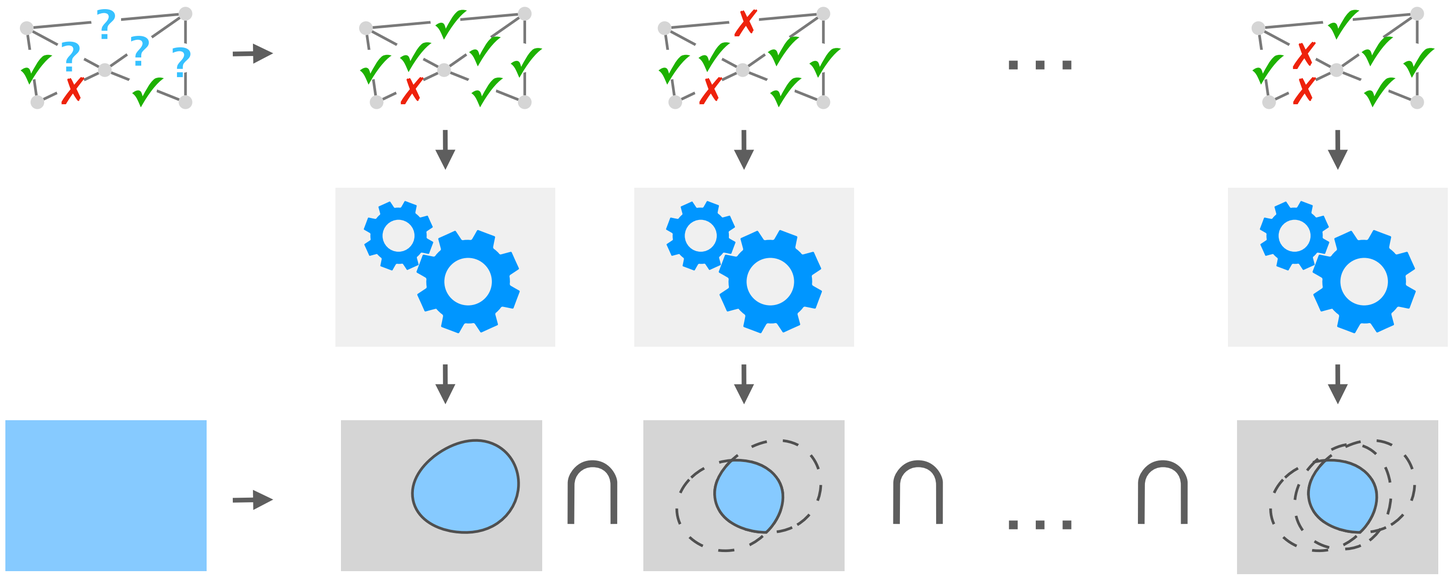
Config2Spec automatically mines the network's full specification from its configuration and the given failure model



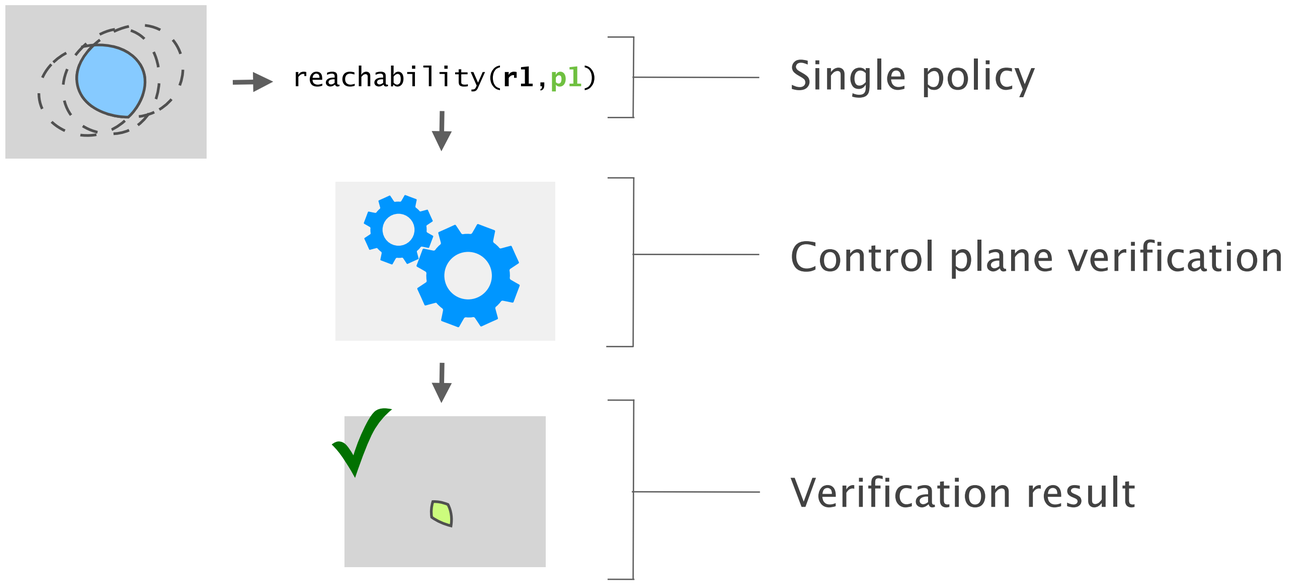
Data plane analysis provides us with a guess of the specification



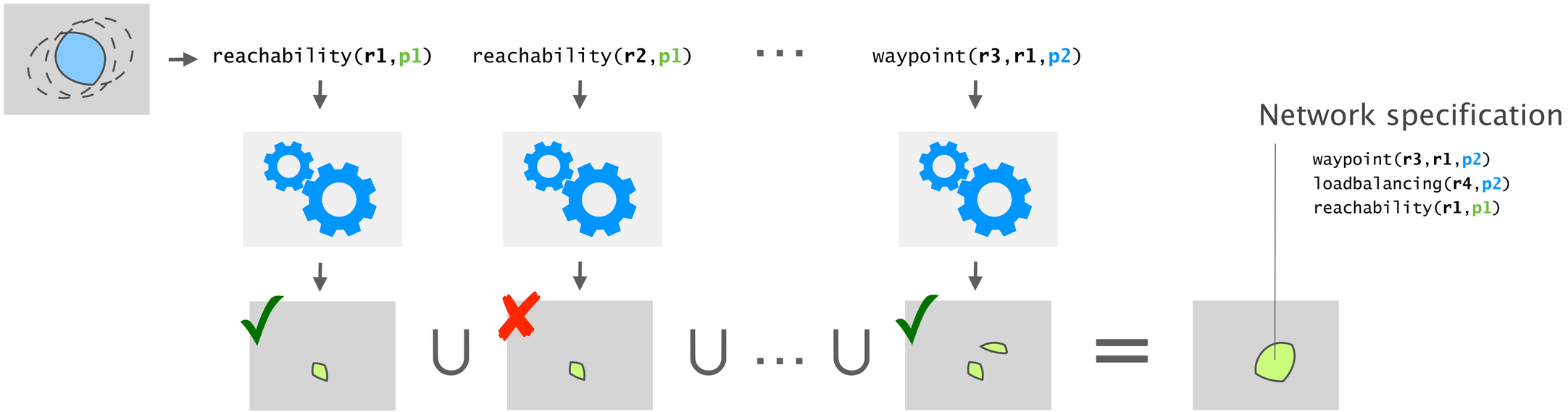
Data plane analysis provides us with a guess of the specification



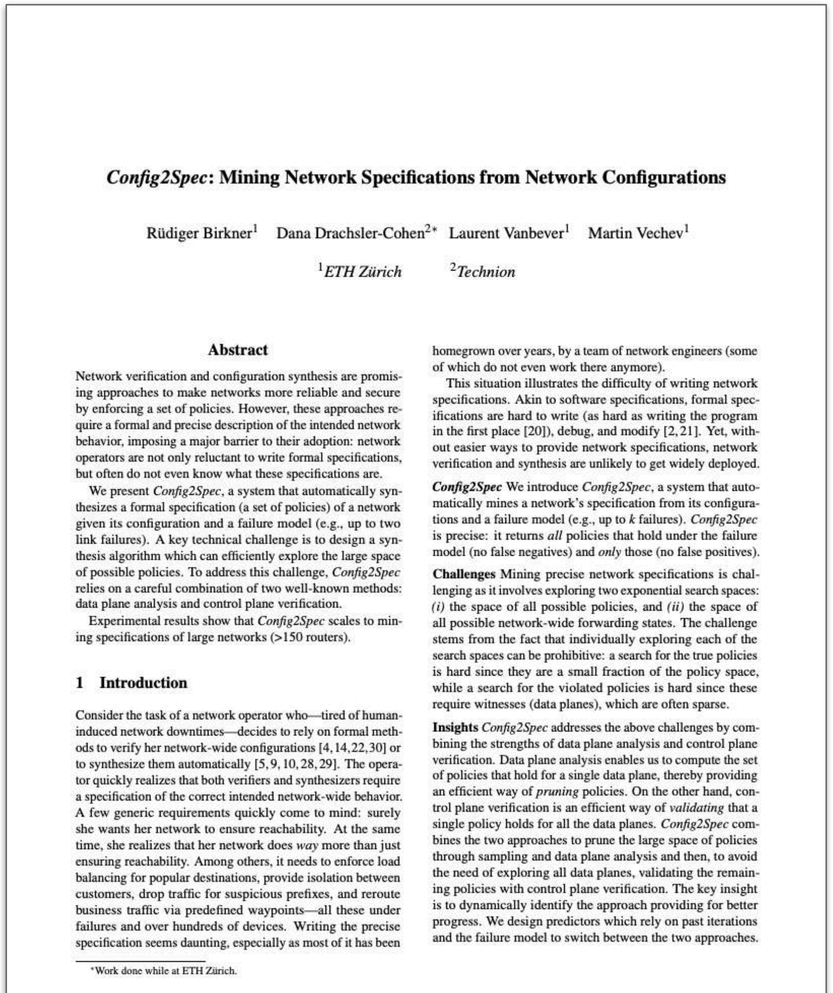
Control plane verification allows us confirm the specification guess one-by-one



Control plane verification allows us confirm the specification guess one-by-one



Config2Spec:
Mining Network Specifications from Network Configurations



***Config2Spec*: Mining Network Specifications from Network Configurations**

Rüdiger Birkner[1] Dana Drachsler-Cohen[2*] Laurent Vanbever[1] Martin Vechev[1]

[1]*ETH Zürich* [2]*Technion*

Abstract

Network verification and configuration synthesis are promising approaches to make networks more reliable and secure by enforcing a set of policies. However, these approaches require a formal and precise description of the intended network behavior, imposing a major barrier to their adoption: network operators are not only reluctant to write formal specifications, but often do not even know what these specifications are.

We present *Config2Spec*, a system that automatically synthesizes a formal specification (a set of policies) of a network given its configuration and a failure model (e.g., up to two link failures). A key technical challenge is to design a synthesis algorithm which can efficiently explore the large space of possible policies. To address this challenge, *Config2Spec* relies on a careful combination of two well-known methods: data plane analysis and control plane verification.

Experimental results show that *Config2Spec* scales to mining specifications of large networks (>150 routers).

1 Introduction

Consider the task of a network operator who—tired of human-induced network downtimes—decides to rely on formal methods to verify her network-wide configurations [4, 14, 22, 30] or to synthesize them automatically [5, 9, 10, 28, 29]. The operator quickly realizes that both verifiers and synthesizers require a specification of the correct intended network-wide behavior. A few generic requirements quickly come to mind: surely she wants her network to ensure reachability. At the same time, she realizes that her network does *way* more than just ensuring reachability. Among others, it needs to enforce load balancing for popular destinations, provide isolation between customers, drop traffic for suspicious prefixes, and reroute business traffic via predefined waypoints—all these under failures and over hundreds of devices. Writing the precise specification seems daunting, especially as most of it has been homegrown over years, by a team of network engineers (some of which do not even work there anymore).

This situation illustrates the difficulty of writing network specifications. Akin to software specifications, formal specifications are hard to write (as hard as writing the program in the first place [20]), debug, and modify [2, 21]. Yet, without easier ways to provide network specifications, network verification and synthesis are unlikely to get widely deployed.

Config2Spec We introduce *Config2Spec*, a system that automatically mines a network's specification from its configurations and a failure model (e.g., up to k failures). *Config2Spec* is precise: it returns *all* policies that hold under the failure model (no false negatives) and *only* those (no false positives).

Challenges Mining precise network specifications is challenging as it involves exploring two exponential search spaces: *(i)* the space of all possible policies, and *(ii)* the space of all possible network-wide forwarding states. The challenge stems from the fact that individually exploring each of the search spaces can be prohibitive: a search for the true policies is hard since they are a small fraction of the policy space, while a search for the violated policies is hard since these require witnesses (data planes), which are often sparse.

Insights *Config2Spec* addresses the above challenges by combining the strengths of data plane analysis and control plane verification. Data plane analysis enables us to compute the set of policies that hold for a single data plane, thereby providing an efficient way of *pruning* policies. On the other hand, control plane verification is an efficient way of *validating* that a single policy holds for all the data planes. *Config2Spec* combines the two approaches to prune the large space of policies through sampling and data plane analysis and then, to avoid the need of exploring all data planes, validating the remaining policies with control plane verification. The key insight is to dynamically identify the approach providing for better progress. We design predictors which rely on past iterations and the failure model to switch between the two approaches.

*Work done while at ETH Zürich.

Check our NSDI'20 paper and talk

as there is much more behind Config2Spec

We are still improving Config2Spec through

richer specifications and automatic bug detection

Please reach out to us at rbirkner@ethz.ch

with all your inputs and feedback

nsg.ee.ethz.ch

Config2Spec

Mining Network Specifications from Network Configurations

Rüdiger Birkner

Dana Drachsler-Cohen

Martin Vechev

Laurent Vanbever

nsg.ee.ethz.ch