Predicting Specification Violations During BGP Convergence

Predicting Specification Violations During BGP Convergence

Roland Schmid
ETH Zurich
roschmi@ethz.ch

Tibor Schneider
ETH Zurich
sctibor@ethz.ch

Georgia Fragkouli
ETH Zurich
gfragkouli@ethz.ch

Laurent Vanbever
ETH Zurich
lvanbever@ethz.ch

ABSTRACT

Analyzing a network's behavior during convergence is challenging due to its highly non-deterministic nature. To address this, we developed BGPSeer, the first analyzer that predicts specification violations during BGP convergence without network disruption.

To do this both accurately and fast, BGPSeer builds a probabilistic network timing model based on hardware measurements that allows to sample BGP message orderings, from which BGPSeer estimates violation times. We implemented BGPSeer by extending an open-source BGP simulator and show that it achieves 85-99% accuracy in estimating violation times in less than ten seconds.

CCS CONCEPTS

• **Networks** → **Protocol testing and verification**; *Routing protocols*; *Network reliability*; *Network simulations*; *Formal specifications.*

ACM Reference Format:
Roland Schmid, Tibor Schneider, Georgia Fragkouli, and Laurent Vanbever. 2023. Predicting Specification Violations During BGP Convergence. In *Proceedings of the CoNEXT Student Workshop 2023 (CoNEXT-SW '23), December 8, 2023, Paris, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3630202.3630235

1 INTRODUCTION

Applications increasingly demand better performance, driving network operators to optimize their networks to meet competitive specifications. Ultimately, this will require that networks not only guarantee properties in the stable state but also *during BGP convergence.* While recent frameworks prevent transient violations caused by planned reconfigurations [3], they fundamentally cannot handle violations caused by *external* routing events, which occur frequently [1]. Thus, network operators need a tool that, given a network's topology, configuration, and a routing event, accurately estimates which specification violations to expect and how long these last, i.e., the so-called *transient violation times.*

Prior work cannot estimate violation times without disrupting the network. Measurement approaches measure violations on live networks and require that the operator deploys each configuration and injects each routing event separately. This process, however, is time-consuming and revenue-damaging, as it causes the violations that the operator wants to avoid. Also, state-of-the-art verification tools [2] are limited to the stable state and fundamentally have no notion of time, which is required to estimate violation times.

CoNEXT-SW '23, December 8, 2023, Paris, France

ACM ISBN 979-8-4007-0452-9/23/12.
https://doi.org/10.1145/3630202.3630235

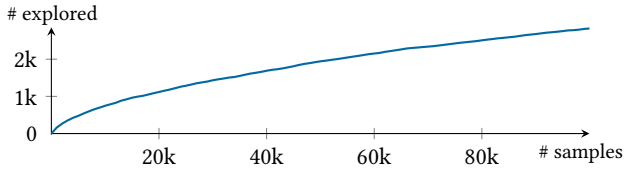

Figure 1: BGPSeer explores >2'000 distinct series of forwarding states, even when ignoring update times.

To sum up, an important research question remains open: *Is it possible to build an analyzer that predicts transient specification violations caused by external routing events?*

We argue that this is possible and propose BGPSeer, the first tool that reasons about the transient violation times for generic networks. The main challenge is modeling the non-determinism inherent to the BGP convergence process, which results in multiple message orderings to consider and no clear way to rank those orderings or sample the ones that are likely to occur in practice. Indeed, Fig. 1 shows that BGPSeer finds >2'000 plausible series of forwarding states that a network can experience during convergence. Our key insight is that we can model the complex BGP convergence process by modeling the routers' reaction times to BGP messages.

BGPSeer uses a hardware testbed to measure precise distributions of control- and data-plane reaction times. These are used for sampling realistic convergence behavior and inferring the violation times accordingly. Preliminary evaluation on a real network topology shows that BGPSeer predicts violation times within a few seconds and with 85-99% accuracy compared to measurements taken from a hardware testbed.

2 THE BGPSEER APPROACH

BGPSeer estimates transient specification violations in two phases: (i) accurately modeling routers' convergence behavior with a timing model and augmenting an existing control plane simulator to produce a time-aware BGP simulator; and (ii) sampling realistic message orderings and inferring transient violation times from them. In the following, we describe these two phases in more detail.

Obtaining a transient BGP simulator. First, we use measurements from a hardware testbed to gather a precise router-local timing model. This model consists of distributions of control- and data-plane *reaction times* in response to BGP messages. Instead of fixed-value estimates, we use distributions to capture the non-deterministic nature of hardware reactions. The control-plane distribution captures the duration from when a router receives a BGP message to when the router sends responses triggered by this message to its peers. Similarly, the data-plane distribution captures the duration from when a router receives a BGP message to when its local changes take effect in the data plane.

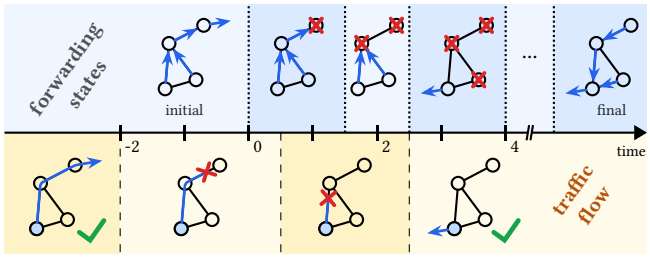

Figure 2: BGPSEER deduces violation times by sampling many series of forwarding states that may occur during convergence and relating these to the traffic flow.

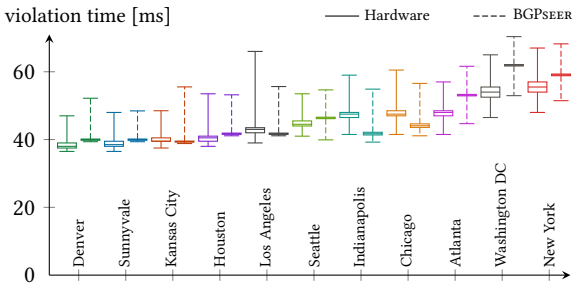

Figure 3: BGPSEER estimates the violation times with 85-99% accuracy compared to hardware measurements.

Next, BGPSEER produces a time-aware BGP simulator: When a router receives a BGP message, it queries the control-plane simulator to determine the control-/data-plane actions triggered by this message. BGPSEER estimates the time it would take for these actions to take effect using the router-local timing model and incorporating both the load level at the router, and the propagation delays.

Sampling violation times. We say that the *violation time* at router r is the cumulative timespan during which traffic emitted at r does not reach its destination. BGPSEER uses the above transient BGP simulator to sample realistic message orderings and timings. Effectively, this results in every run of BGPSEER sampling a time series of forwarding states. For example, the upper part of Fig. 2 illustrates such a time series following a prefix withdrawal, where the propagation delay is Δt=1 time step per hop and the control- and data-plane reactions are fixed to Δt=0.5 time steps.

The updates of the forwarding states, however, are not necessarily aligned with their effects on the traffic flow sent from a router r. For example, the bottom part of Fig. 2 shows that traffic originating from r at t=-2 gets dropped already, which is before the prefix withdrawal occurs at t=0. Hence, BGPSEER correlates propagation delays with the forwarding updates: determining all the intervals during which any traffic flow originating at a router is dropped, e.g., due to black holes along the path of this flow. Finally, BGPSEER reports the sum of these intervals' durations as the respective violation times.

3 PRELIMINARY EVALUATION

We evaluate whether BGPSEER predicts violation times accurately and fast, focusing on a single BGP prefix-withdrawal. To evaluate the accuracy, we extend our hardware testbed to introduce BGP events and measure the resulting transient violation times for each individual router. We collect multiple samples for both BGPSEER and the hardware testbed, and compare their respective distributions.

Implementation. We implemented a prototype of BGPSEER in Rust by extending the open-source simulator `bgpsim` [3]. To build the hardware testbed, we use three Cisco Nexus 7000 devices running up to twelve virtual routers and impose artificial propagation delays to emulate a real network topology. We ran BGPSEER on an AMD EPYC 7742 64-Core Processor, clocked at 1.5GHz, using 100 threads and less than 1 GB of RAM.

Methodology. We estimate reachability violation times on the Abilene topology from Topology Zoo, which consists of eleven routers. We connect the routers in an iBGP full mesh and set the link propagation delays based on the geographic location of the connected routers. Two routers, Los Angeles (LA) and Kansas City (KC), learn routes from two different external peers for the same prefix p. Initially, the network is in a stable state where all routers prefer the route through LA. At some point, the external peer at LA withdraws the route for p. This results in LA instantaneously learning the withdrawal and the network starting to reconverge to a new stable state, where all routers use the route through KC. In both stable states, the network does not violate reachability.

Results. Figure 3 shows the transient violation times estimated by each method at each router. We conclude that BGPSEER accurately estimates violation times, as the relative error between BGPSEER's and the testbed's median violation times is always below 15%. Thanks to its perfect parallelizability, BGPSEER collects 100'000 samples per router within 8.2 seconds. In contrast, the testbed takes three orders of magnitude more time to collect only 1'000 samples per router, as it incurs the cost of configuring the physical devices.

4 CONCLUSION AND FUTURE WORK

We have shown that despite the non-determinism of the BGP convergence process, it is feasible to predict transient violation times accurately and fast on a real topology.

In the future, we want to explore the accuracy of BGPSEER across diverse network scenarios. We expect this to depend mainly on the expressiveness of the timing model, which is currently tailored to single-prefix withdrawals. Can the model capture other types of BGP messages, e.g., 1'000-prefix withdrawals caused by a failing eBGP session? Can it model and scale to larger networks? Support complex route-maps? Generalize to other hardware?

ACKNOWLEDGMENTS

The research leading to these results was supported by an ERC Starting Grant (SyNET) 851809.

REFERENCES

[1] The BGP Instability Report. https://bgpupdates.potaroo.net/instability/bgpupd.html. 2023.
[2] R. Beckett, A. Gupta, R. Mahajan, and D. Walker. A General Approach to Network Configuration Verification. In *Proc. ACM SIGCOMM.* 2017.
[3] T. Schneider, R. Schmid, S. Vissicchio, and L. Vanbever. Taming the Transient While Reconfiguring BGP. In *Proc. ACM SIGCOMM.* 2023.