Crescent: Emulating Heterogeneous Production Network at Scale

Zhaoyu Gao, Anubhavindhi Abhashkumar, Zhen Sun, Weirong Jiang, Yi Wang
Outline

- Background & Motivation
- Challenges
- Proposed Solution: Crescent
- Evaluation Result
- Future Work & Summary
Background: ByteDance’s Network
Motivation

- ByteDance’s network scale increases steadily.
- Number of network changes increases much more rapidly.
- Incidents caused by network changes also happened more frequently since 2020-1H.

The trend of network incidents in 2020
Incident Example

Regional WAN

Global WAN

AS path prepend:
[asn₁, asn₂, asn₃] -> [asn₁, asn₂, asn₃, asn₃]
Lesson Learnt from Past Incidents

Vendor-specific behaviors (VSBs) are hard to prevent because of unawareness of VSBs.

Emulating only the devices under test (DUTs) is insufficient to catch the impact of a change.
Potential Solutions: Simulation vs Emulation

CPV (Control Plane Verification)
- Batfish [1]
- Hoyan [2]
- ...

Emulation
- EVE-NG [3]
- GNS3 [4]
- Vrnetlab [5]
- CrystalNet [6]
- ...

Hard to catch VSBs

Unable to find a safe static emulation boundary.


[3] https://www.eve-ng.net/
Challenges


2. Scalability: large testbed creation over a distributed setup.


How to emulate a large-scale network effectively with limited resources?
Incident Analysis

For network incidents in the past 3 years

- 1/3 incidents were caused by network changes (configuration and topology updates).
- 30% of these incidents involved VSBs (vendor-specific behaviors).
- ~50% network changes are applied to core devices, while over 90% of network incidents happened on core devices.
Observation: Network Symmetry

- High standardization (topology and configuration) on DCN non-core devices.

- Topology and configuration on core devices can not be highly standardized.
Proposed Solution: Crescent

Challenge 1: Cost v.s. Coverage
Canary testbed: a long-time running testbed with all core devices and selected non-core devices.

Challenge 2: Scalability
Partitioning algorithm to scale out canary testbed & connecting DUTs to canary.

Challenge 3: Efficient Verification
Automated monitoring and verification tools.
Cross-host link creation overhead is much higher than same-host link creation.
Crescent – Partitioning Algorithm

Goal: Minimize # of cross-host links

NP-hard problem

Solution:
A variant of community detection algorithm

Implemented with a heuristic greedy algorithm

\[
\begin{align*}
\text{min} & \quad \sum_{i,j \in E_{ij}} w_e \\
\text{s.t.} & \quad 1 \leq i, j \leq n \\
E_{ij} & = E \cap V_i \times V_j \\
V & = V_1 \cup V_2 \cup \ldots \cup V_n \\
V_i \cap V_j & = \emptyset \\
\sum_{v \in V_k} w_v & \leq C, 1 \leq k \leq n
\end{align*}
\]
Crescent – Connecting DUTs to Canary

Expansion: find paths from DUTs to canary.

Connection: dynamically connect DUTs to canary.

1. Connect DUTs to canary.
2. Execute MOPs and verify.
Crescent – Automated Monitoring and Verification

Monitoring tools:
- Pingmesh [7]
- route differ
- config checker

Verification tools: homebrew DPV

Evaluation: Partitioning Schemes

Node-to-host assignment schemes:
- Crescent: a partitioning scheme generated by Crescent partitioning algorithm proposed in this work.
- Geo-manual: a partitioning scheme by a network expert manually partitioning our network based on geographical affinity.
- Random: a partitioning scheme randomly assigning nodes to hosts.
Evaluation Results

Connection Time

Bootup Time

- Crescent
- Geo-manual
- Random
In-Production Deployment Result

Most commonly-detected errors: typos

Some incidents can still be missed.

Used beyond network change, e.g., SDN controller test.
Future Work

- Tighten the boundary: some core devices may not be needed in canary.

- Shrink the scale: route injection.

- Train CPV: use emulation to generate ground truth to feed to CPV.
Summary

- Network changes are a major source of network incidents.

- We propose Crescent, a large-scale high-fidelity emulation platform containing all core devices combined with timely verification.

- To achieve high scalability, we use a multihost setup and a partitioning algorithm for a scalable node-to-host assignment to reduce the number of cross-host links and to minimize bootup time and connection time.

- Our in-production deployment shows that Crescent helped reduce change-induced network incidents.
THANKS