Crescent: Emulating Heterogeneous Production Network at Scale

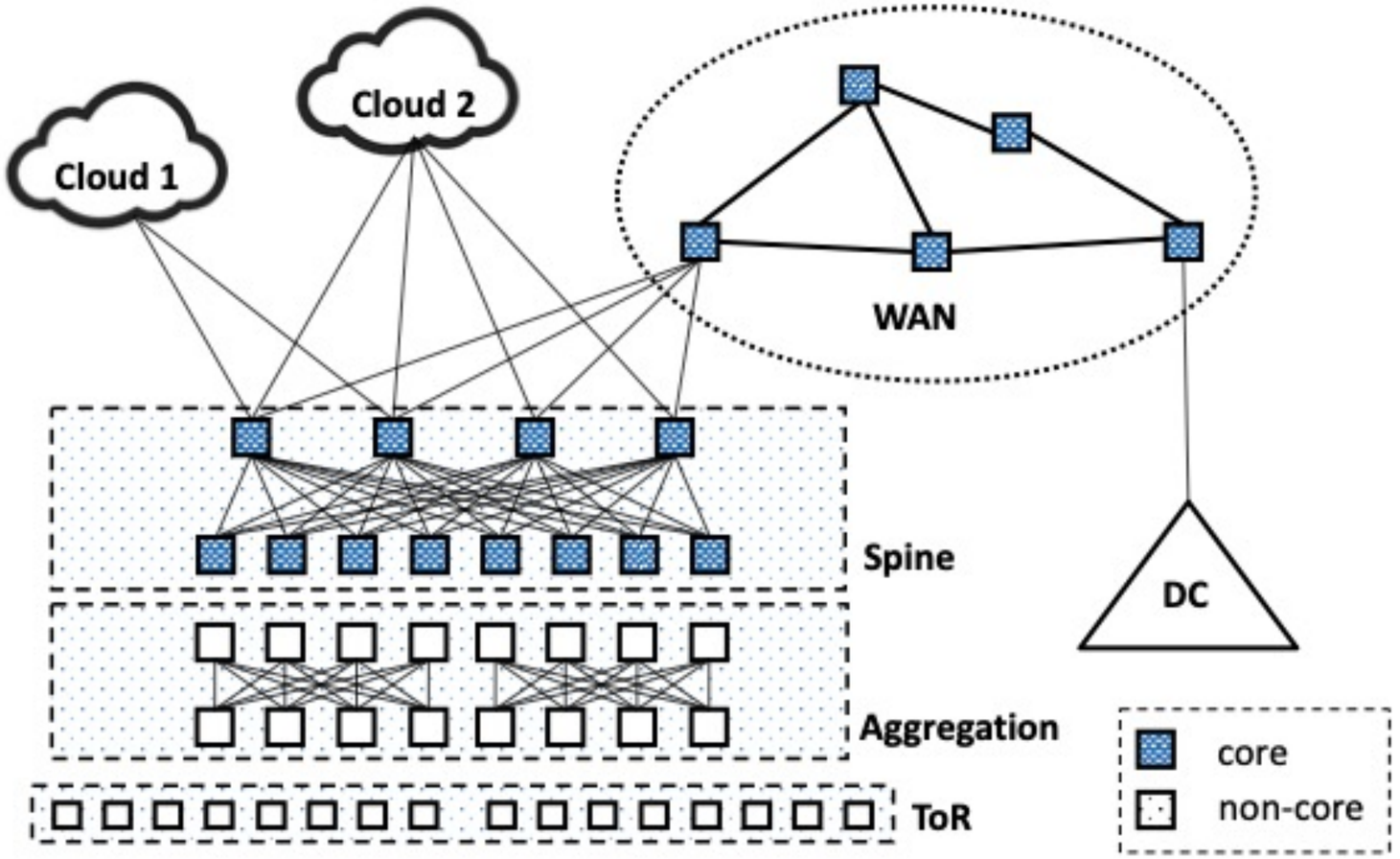
Zhaoyu Gao, Anubhavnidhi Abhashkumar, Zhen Sun,
Weirong Jiang, Yi Wang

ByteDance 字节跳动

Outline

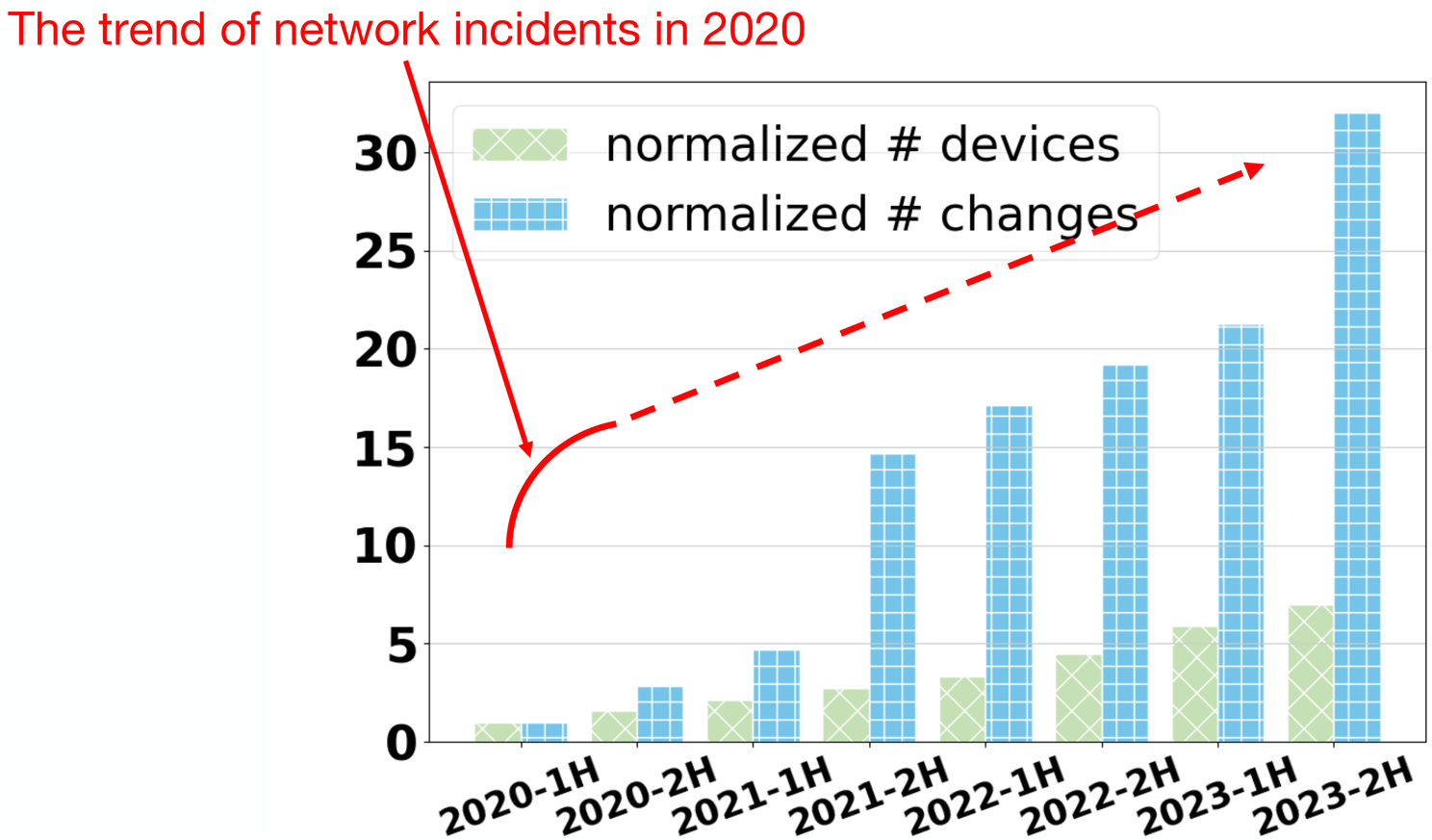
- Background & Motivation
- Challenges
- Proposed Solution: Crescent
- Evaluation Result
- Future Work & Summary

Background: ByteDance's Network

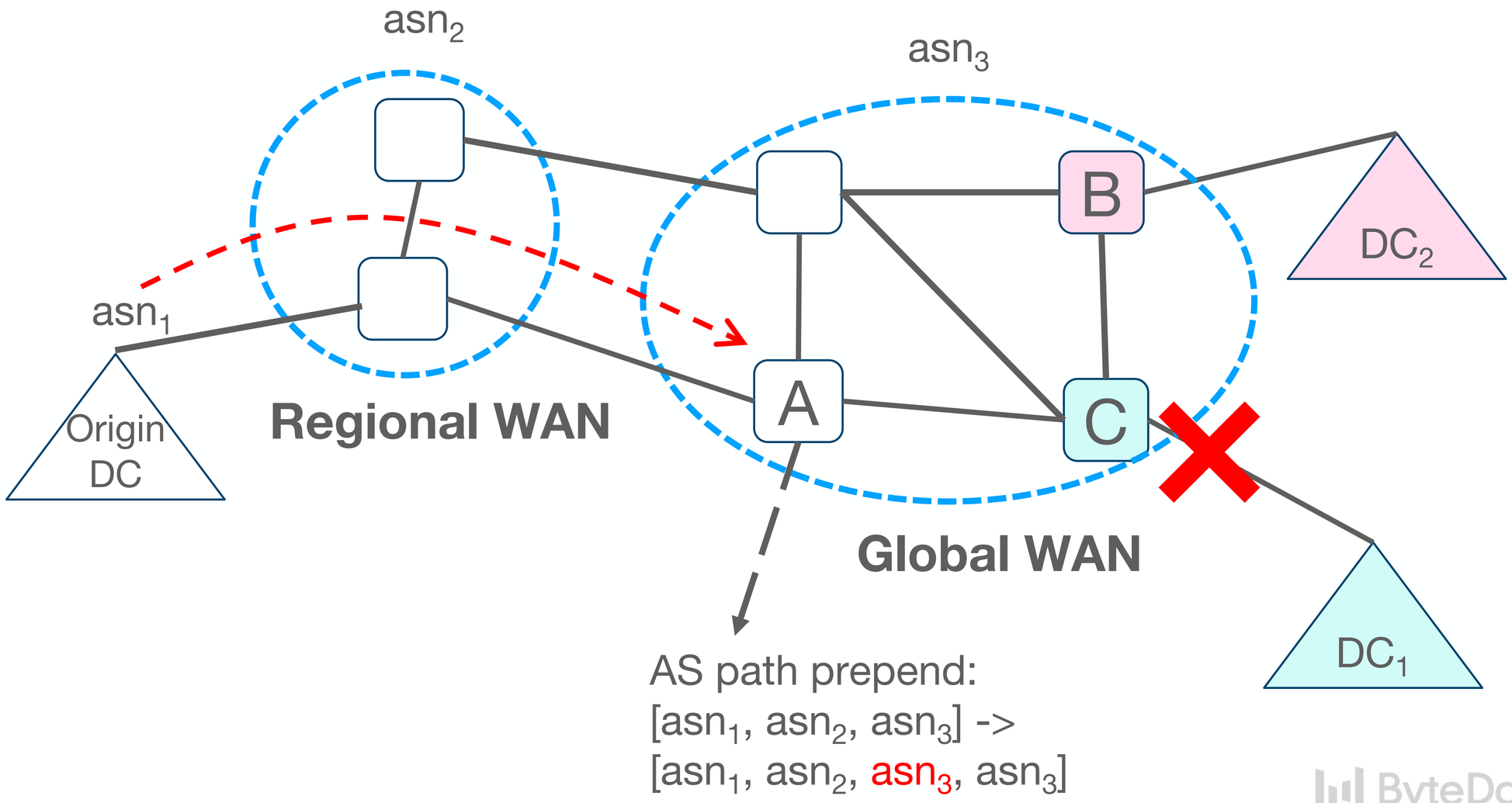


Motivation

- 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.



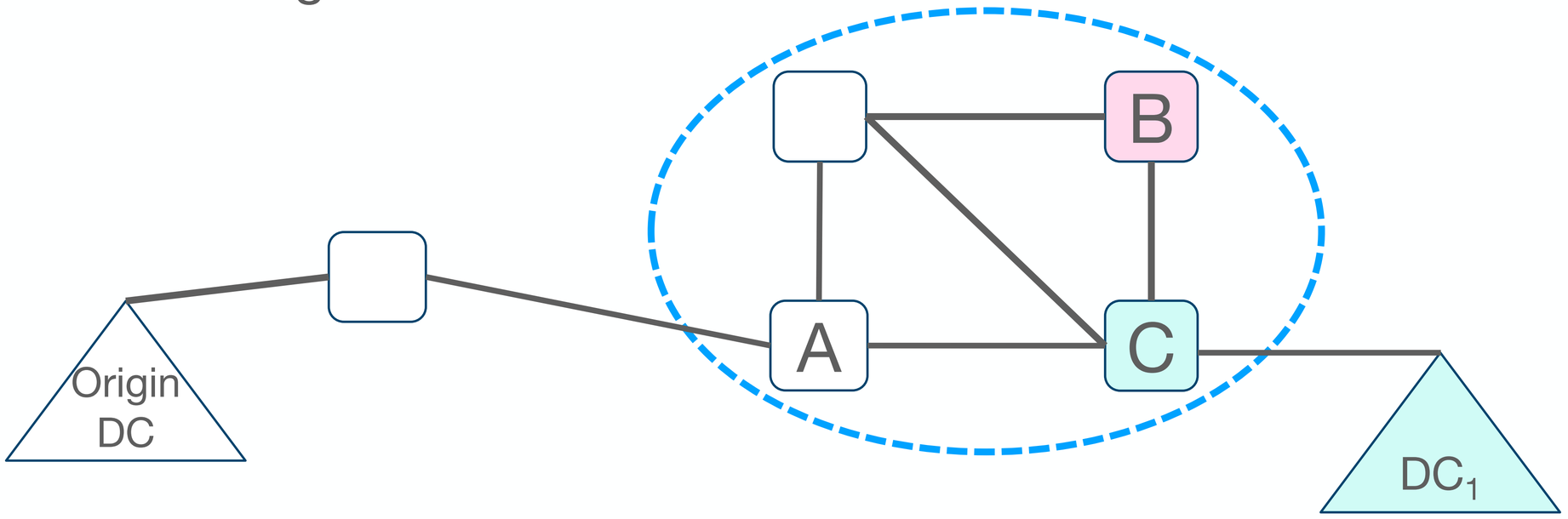
Incident Example



Lesson Learnt from Past Incidents

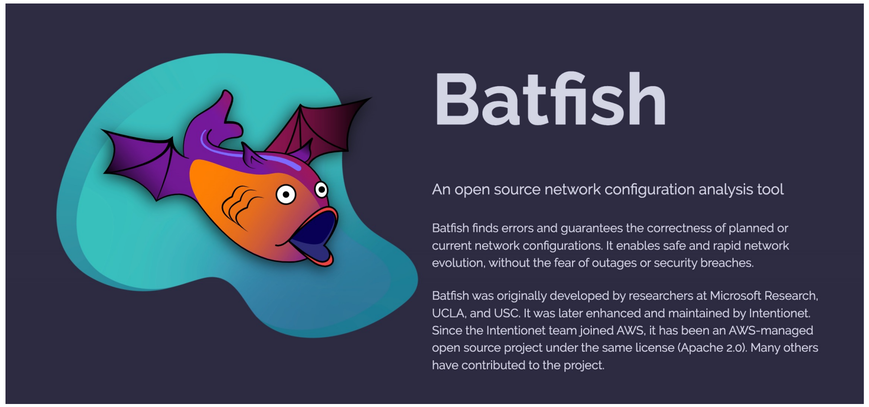
Vendor-specific behaviors (VSBs) are hard to prevent because of unawareness of VSBs.

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]
- …



Hard to catch VSBs

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

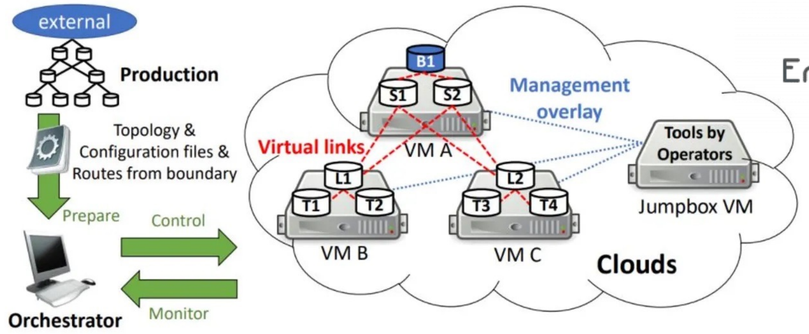


Figure 1: The architecture of CrystalNet

Hard to scale out

Unable to find a safe static emulation boundary.

[1] Matt Brown, Ari Fogel, Daniel Halperin, Victor Heo- rhiadi, Ratul Mahajan, and Todd Millstein. **Lessons from the evolution of the batfish configuration analysis tool.** In *Proceedings of the 2023 Conference of the ACM Special Interest Group on Data Communication*, pages 122–135, 2023.

[2] Fangdan Ye, Da Yu, et al**. Accuracy, scalability, coverage: A practical configuration verifier on a global wan**. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 599–614, 2020

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

[4] https://www.gns3.com/

[5] https://github.com/vrnetlab/vrnetlab

[6] Hongqiang Harry Liu, Yibo Zhu, Jitu Padhye, Jiaxin Cao, Sri Tallapragada, Nuno P Lopes, Andrey Ry- balchenko

Challenges

1. Cost v.s. Coverage: hard to predict blast radius of network change.

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

3. Efficient Verification: emulation alone can not verify network at scale.

How to emulate a large-scale network effectively with limited resource?

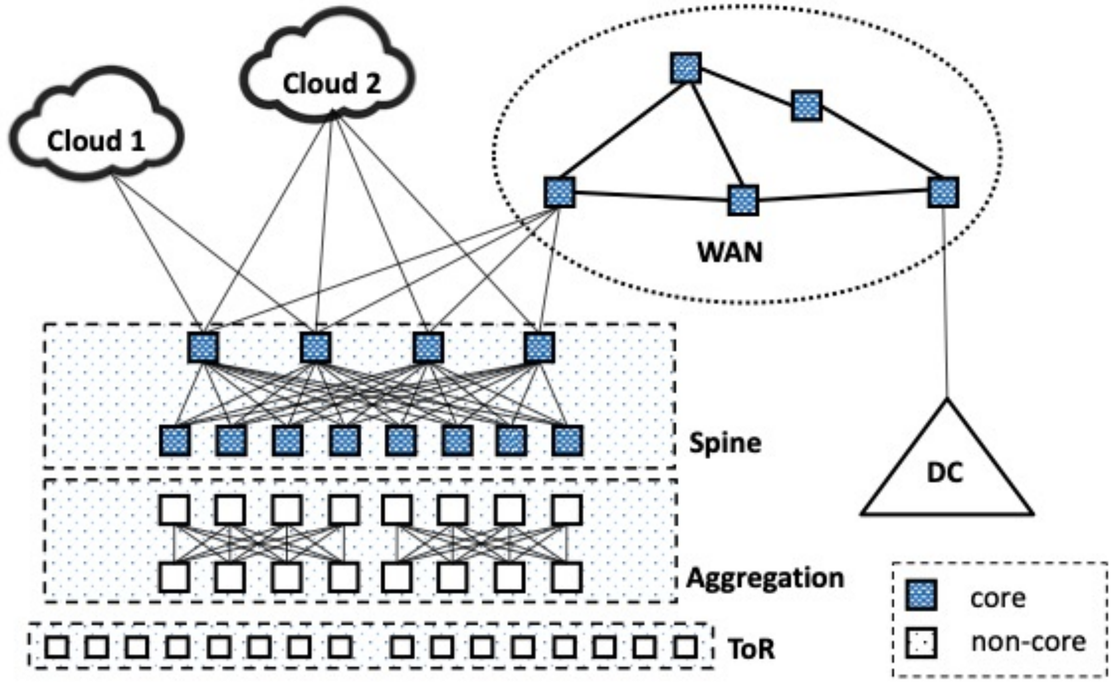
Incident Analysis

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

Challenge1: Cost v.s. Coverage

Canary testbed: a long-time running testbed with all core devices and selected non-core devices.

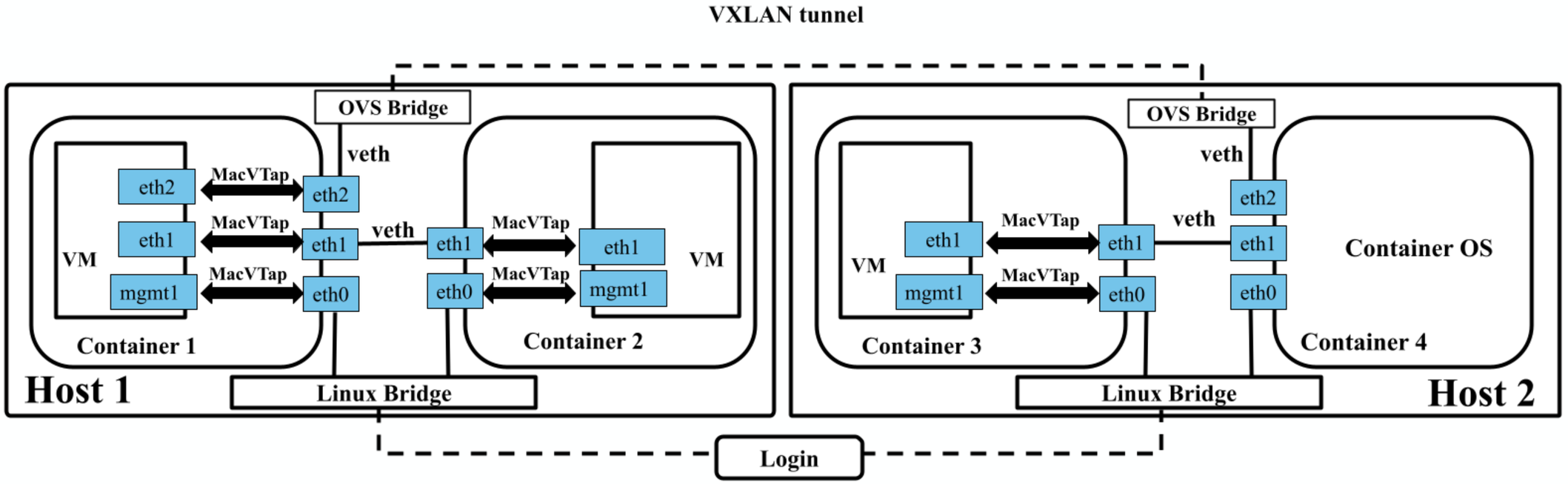
Challenge 2: Scalability

Partitioning algorithm to scale out canary testbed & connecting DUTs to canary.

Challenge 3: Efficient Verification

Automated monitoring and verification tools.

Crescent - Implementation



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

$$\min \quad \sum_{i,j} \sum_{e \in E_{ij}} w_e$$

$$\text{s.t.} \quad 1 \le i, j \le n$$

$$E_{ij} = E \cap V_i \times V_j$$

$$V = V_1 \cup V_2 \cup \dots V_n$$

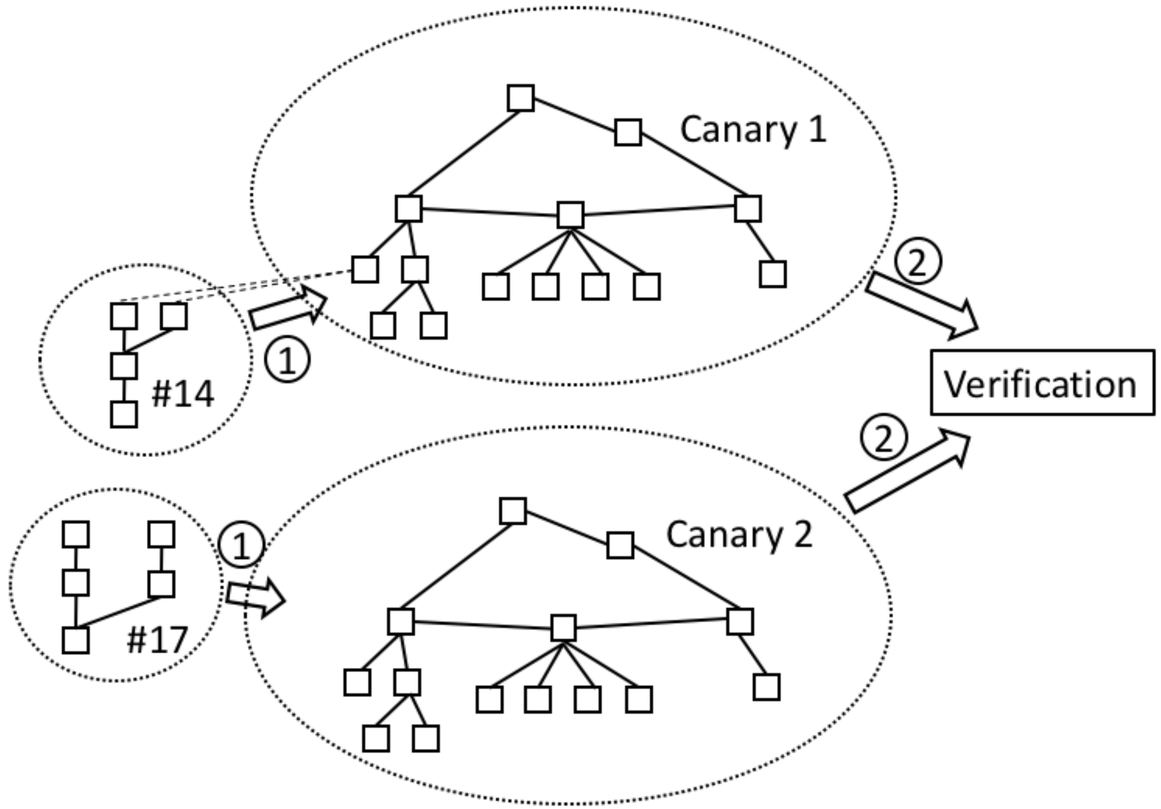
$$V_i \cap V_j = \emptyset$$

$$\sum_{v \in V_k} w_v \le C, 1 \le k \le n$$

Crescent – Connecting DUTs to Canary

Expansion: find paths from DUTs to canary.

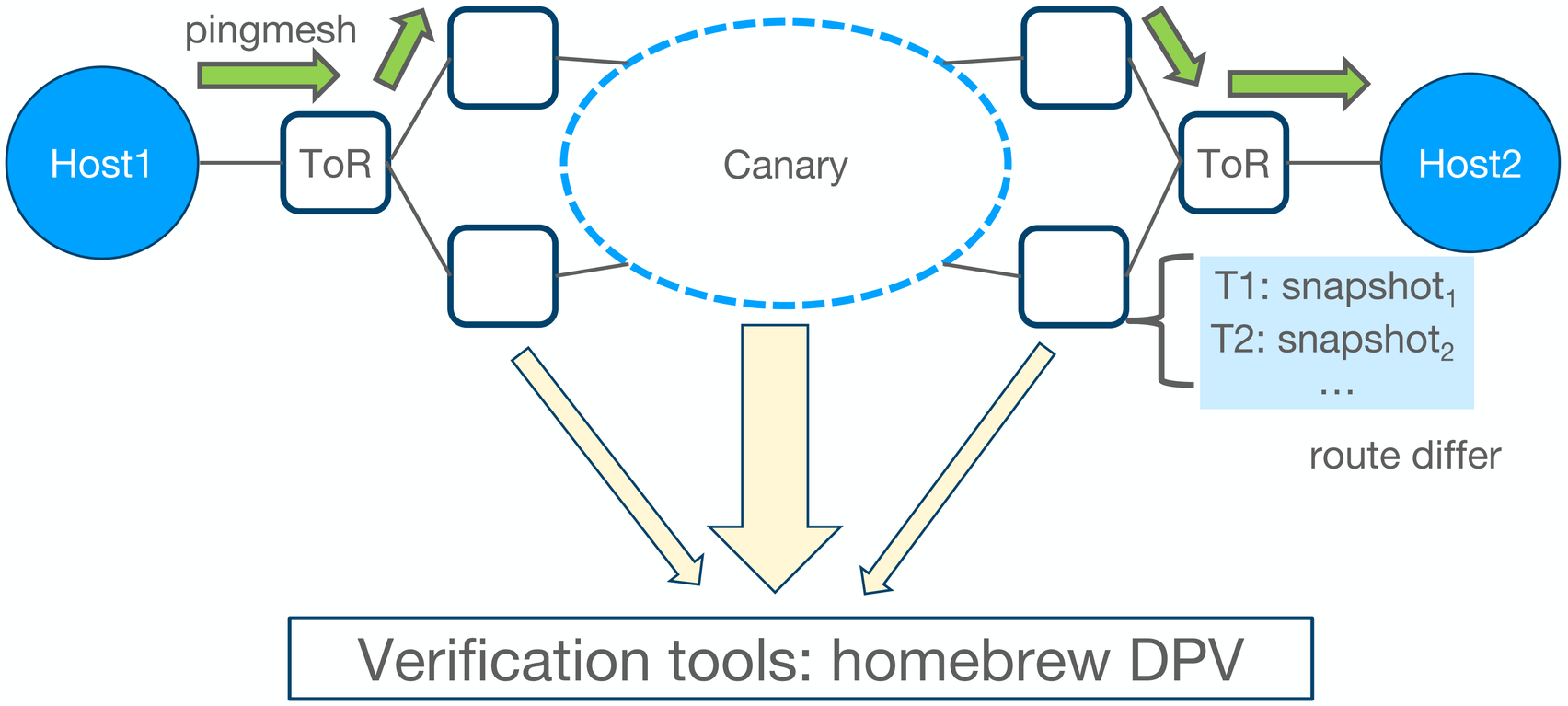
Connection: dynamically connect DUTs to canary.



1. Connect DUTs to canary.
2. Execute MOPs and verify.

Crescent – Automated Monitoring and Verification

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



[7] Chuanxiong Guo, Lihua Yuan, Dong Xiang, Yingnong Dang, Ray Huang, Dave Maltz, Zhaoyi Liu, Vin Wang, Bin Pang, Hua Chen, et al. **Pingmesh: A large-scale system for data center network latency measurement and analysis**. In *Proceedings of the 2015 Conference of the ACM Special Interest Group on Data Communication*, pages 139–152, 2015.

Evaluation: Partitioning Schemes

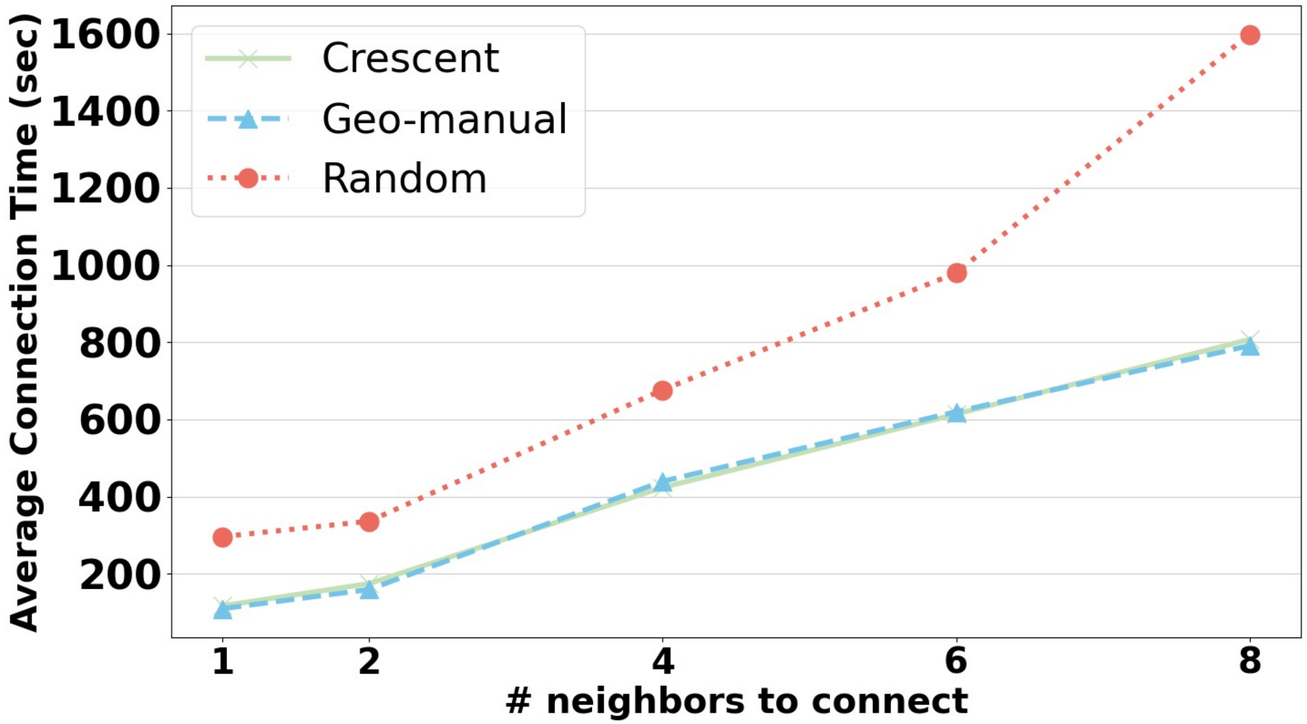
Node-to-host assignment schemes:

- Crescent: a partitioning scheme generated by Crescent partitioning algorithm proposed in this work.

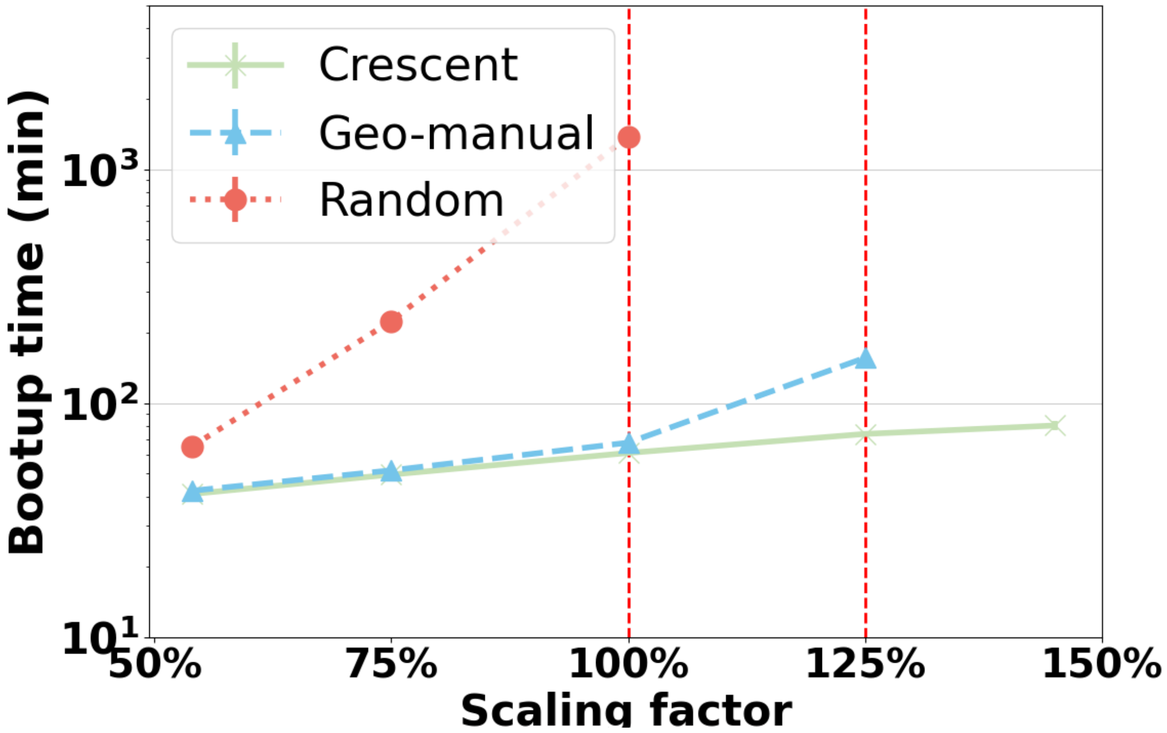
- Geo-manual: a partitioning scheme by a network expert manually partitioning our network based on geographical affinity.

- Random: a partitioning scheme randomly assigning nodes to hosts.

Evaluation Results



Connection Time



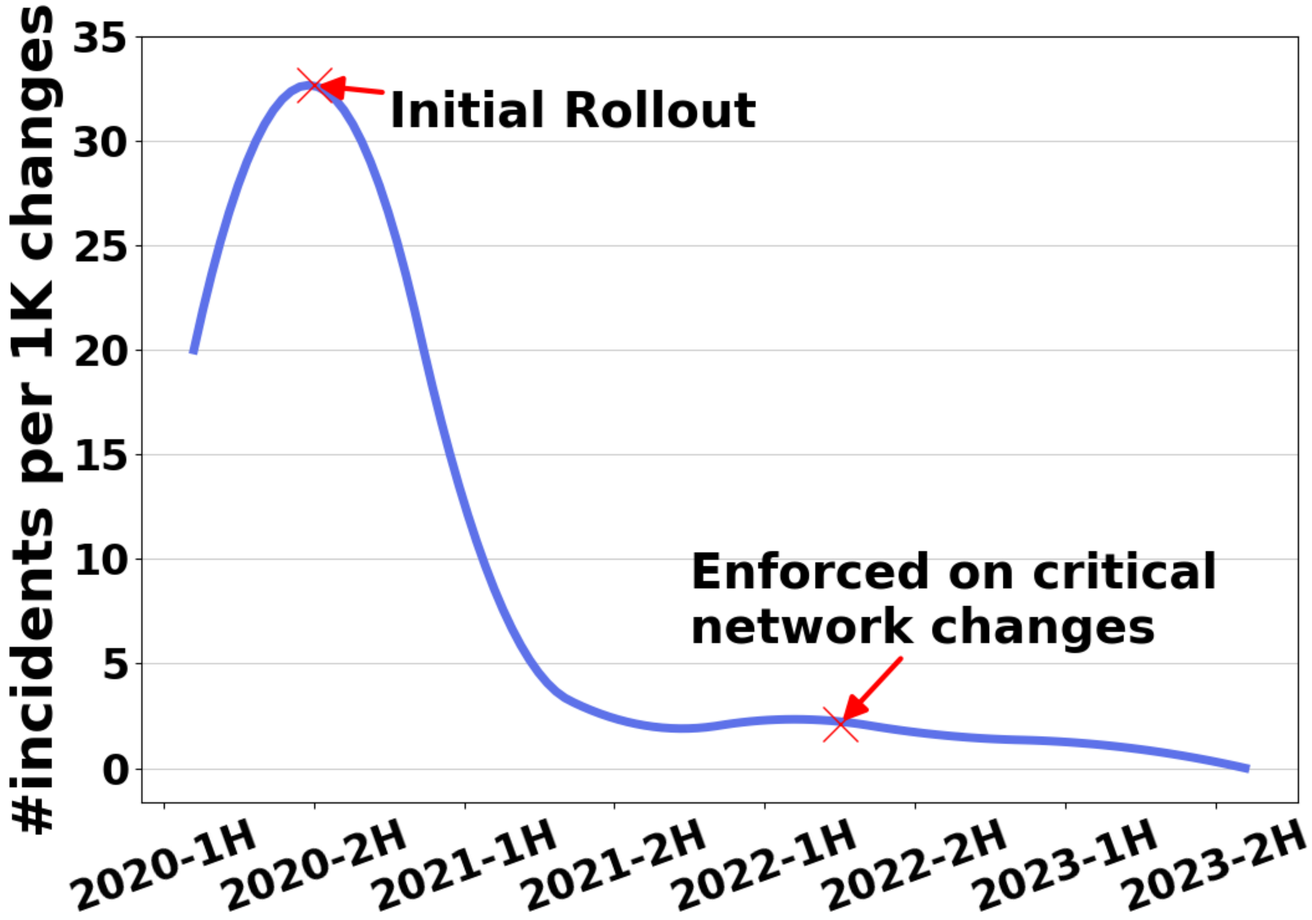
Bootup Time

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