CrystalNet

Faithfully Emulating Large Production Networks

Hongqiang Harry Liu, Yibo Zhu
Nuno Lopes, Andrey Rybalchenko
Microsoft Research

Jitu Padhye, Jiaxin Cao, Sri Tallapragada,
Guohan Lu, Lihua Yuan
Microsoft Azure
Reliability is vital to cloud customers
However, cloud reliability remains elusive

Global DNS outage hits Microsoft Azure customers
Update: An Azure DNS outage, which affected Microsoft customers and services across the globe for several hours, now seems to be mostly mitigated.

Microsoft confirms Azure outage was human error

Data Center • Cloud
Google Cloud rolls back changes after 18-hour load balancer brownout
VMs across US, Europe and Asia all unable to “connect to backends”
Cloud downtime cost:
- 80% reported $50k/hour or above
- 25% reported $500k/hour or above

- USA Today Survey of 200 data center managers

Cloud availability requirement:
- 82% require 99.9% (3 nines) or above
- 42% require 99.99% (4 nines) or above
- 12% require 99.999% (5 nines) or above

- High availability survey over 100 companies
  by Information Technology and Intelligence Corp
What caused these outages?
Network is a major root cause of outages

<table>
<thead>
<tr>
<th>Cloud A</th>
<th>Cloud B</th>
<th>Cloud C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Date</strong></td>
<td>Aug 22\textsuperscript{nd}, 2017</td>
<td>Sep 20\textsuperscript{th}, 2017</td>
</tr>
<tr>
<td><strong>Impact</strong></td>
<td>Service Fabric, SQL DB, IoT Hub, HDInsight, etc.</td>
<td>DynamoDB service disruption in US-East</td>
</tr>
<tr>
<td><strong>Root Cause</strong></td>
<td>An incorrect network configuration change.</td>
<td>Configuration error during network upgrades.</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>99.99% (four nines)</td>
<td>99.999% (five nines)</td>
</tr>
<tr>
<td><strong>Maximum Downtime per Year</strong></td>
<td>52.56 minutes</td>
<td>5.26 minutes</td>
</tr>
<tr>
<td><strong>Down Time</strong></td>
<td>2 hours</td>
<td>3.5 hours</td>
</tr>
</tbody>
</table>

We must prevent such outages proactively!
We do test network hardware and software

management software

switch configuration

Unit Tests

Feature Tests

Testbeds

Vendor Tests

But, not enough

These tests say little about how they work in production
• Software works differently at different scales
• Software/hardware bugs in corner cases
• ...

7
Root causes of Azure’s network incidents

- **Software Bugs**: 36%
  - Bugs in routers, middleboxes, and management tools
- **Configuration Bugs**: 27%
  - ASIC driver failures, silent packet drops, fiber cuts, power failures
  - wrong ACL policies, route leaking, route blackholes
- **Human Errors**: 6%
  - typos, design flaws
- **Hardware Failures**: 29%
  - Unidentified: 2%

*Data Interval: 01/2015 – 01/2017*
Ideal tests: network validation on **actual production**
configuration + **software** + **hardware** + **topology**
Copying production network is infeasible

![Diagram showing the infeasibility of copying production network due to high cost of hardware.](image)

Most cost is from hardware.
High-fidelity production environments

CUSTOMER IMPACTING NETWORK INCIDENTS IN AZURE (2015-2017)

- Software Bugs
- Configuration Bugs
- Human Errors
- Hardware Failures
- Unidentified

>69%
CrystalNet

A high-fidelity, cloud-scale network emulator
Overview of CrystalNet

Production

Prepare

Control

Monitor

Management VM (Linux, Windows, etc.)

Tools by Operators

Virtual links

Orchestrator

external

topo
config
software version
route

Probing & testing traffic
Challenges to realize CrystalNet

• scalability to emulate large networks

• flexibility to accommodate heterogeneous switches

• correctness and cost efficiency of emulation boundary
Emulation must scale out to multiple servers

1 CPU Core  \times  5000 = 5000 CPU Cores

You need cloud to emulate a cloud network!
Emulation can cross cloud boundary

Public Cloud

load balancer

Internet

Private Cloud

special hardware

private network
Challenges to realize CrystalNet

- **scalability** to emulate large networks
  - scaling out emulations transparently on multiple hosts and clouds

- **flexibility** to accommodate heterogeneous switches

- **correctness** and **cost efficiency** of emulation boundary
Heterogenous switch software sandboxes

Potential switch sandboxes

Docker Container:
- Efficient
- Supported by all cloud providers

Virtual Machine:
- Several vendors only offer this option

Bare-metal:
- Non-virtualizable devices (e.g. middlebox)
- Needed for hardware integration tests
Management challenges by heterogeneity
Management challenges by heterogeneity
Building a homogenous network layer

Key idea: maintaining network with a homogenous layer of containers
• start a PhyNet container for each switch
• build overlay networks among PhyNet containers
• Managing overlay networks with in PhyNet containers
Challenges to realize CrystalNet

• **scalability** to emulate large networks
  ▪ scaling out emulations transparently on multiple hosts and clouds

• **flexibility** to accommodate heterogeneous devices
  ▪ Introducing a homogeneous PhyNet layer to open and unify network name space of devices

• **correctness** and **cost efficiency** of emulation boundary
A transparent boundary is needed

- We cannot extend the emulation to the whole Internet
  - cost
  - hard to get software or policy beyond our administrative domain

Core Network & Internet (non-emulated)

Data Center Network (emulated)

A transparent boundary:
- No sense of the existence of a boundary
- Behaving identically as real networks
CrystalNet constructs static boundaries

Static speaker devices:
• terminate the topology
• maintain connections with emulated devices
• customizable initial routes to emulation
• no reaction to dynamics inside emulation

Core Network & Internet (non-emulated)

Data Center Network (emulated)

Correctness?
An example of an unsafe boundary
A proven safe boundary

The boundary is a single AS, announcements never return

See paper for proofs and safe boundary for OSPF, IS-IS, etc.
Small boundaries significantly reduce cost

Cost savings from Emulating the entire DC: 96%~98% (See the paper for the algorithm)
Case study
Shifting to regional backbones

**Good news:**
- Significantly better performance for intra-region traffic once the migration is finished

**Bad news:**
- It is difficult to achieve this migration without user impact
The migration is a complex operation

Common policies in Azure’s datacenters:
1. Reachability among the servers are always on;
2. Private IP blocks are never announced out of a datacenter;
3. Special private IP blocks must be announced out of a datacenter;
4. Intra-region traffic needs to go through regional backbone;
5. Inter-region traffic needs to go through core backbone;
6. All IP addresses need to be aggregated before being announced by Spines;
7. None public IP addresses should be announced to Leaves or below;

- Operators have developed software and tools, but they have no way to try them out in realistic setting

[Severity 1: Id XXX]
Date: 10/19/2016
Impact: the entire region X unreachable
Root Cause: Human Typo

[Severity 1: Id YYY]
Date: 10/26/2016
Impact: VM crashes and service failures region Y
Root Cause: Wrong operation order
Zero customer impact with CrystalNet

• **Networks:** two largest datacenters and the core and regional backbone between them.

• **Cost of emulation:** $30/hour ($1000/hour without safe & small boundary design)

• **Bugs found:** 50+, including configuration, management script, switch software and operation errors

• **Potential saving:** 5+ outages

• **Incidents in production:** 0
Performance
CrystalNet starts large scale emulations in minutes

<table>
<thead>
<tr>
<th>#Borders</th>
<th>#Spines</th>
<th>#Leaves</th>
<th>#ToRs</th>
<th>#Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(10)$</td>
<td>$O(100)$</td>
<td>$O(1000)$</td>
<td>$O(1000)$</td>
<td>$O(10M)$</td>
</tr>
</tbody>
</table>

One of the largest DC network in Azure

![Emulation Startup Latency](image)

- **Latency (Minutes)**
  - 500VM/2000Cores
  - 1000VM/4000Cores

**Network Ready** | **Route Ready**
Future work

- How to provide root cause trace back
- How to provide automatic testing
- How to check bugs or performance issues in hardware
- What is the theory to search the minimum boundary
- ...
Conclusion

• Network is a major contributor to clouds’ outages
• We build CrystalNet have to prevent network incidents proactively
  • cloud based scalability
  • flexibility to handle heterogeneous switch sandboxes
  • correctness & cost efficiency of a transparent emulation boundary
• CrystalNet is easy to use and has been used as a mandatory network validation process in Azure
Azure is considering providing CrystalNet as a service
Interested? Contact us!

crystalnet-dev@microsoft.com

At Microsoft, our mission is to empower every person and every organization on the planet to achieve more.