The Zero Touch Network

Bikash Koley
For Google Technical Infrastructure

CNSM 2016
For the past 15 years, Google has been building out the largest cloud infrastructure **on the planet.**
100 Billion searches per month on google.com

Source: Google, 2012
A Global Cloud Network
Google Backbone(s)

Internet facing Backbone, B2:
70+ locations in 33 countries

Global Software Defined Inter-DC Backbone: B4
Operational scale

- 30,000+ circuits in operation
- Many tens of network element roles
- Dozen+ vendors
- 4M lines of configuration files
- ~30K configuration changes per month
- > 8M OIDs collected every 5 minutes
At scale stuff breaks!
The Nines and the Outage Budgets

... for **four 9s** availability?

- **99.99% uptime**
  - **4 minutes per month**

... for **five 9s** availability?

- **99.999% uptime**
  - **24 seconds per month**
Why is high network availability a challenge?

- Velocity of Evolution
- Scale
- Management Complexity
Google’s Network Hardware Evolves Constantly

- **Firehose 1.0**
- **Watchtower**
- **Saturn**
- **Firehose 1.1**
- **4 Post**
- **Jupiter**

Google's Network Hardware Evolves Constantly
As does the Network Software
... driven by ever-evolving products
Network Operation is a tradeoff

Traditional network: pick any two of the three

- reliability
- scale
- efficiency

We want all three!
Lessons learned from a decade of high-availability network design
We analyzed over 100 *Post-mortem* reports written over a 2 year period
What is a Post-mortem?

Carefully curated description of a *previously unseen* failure that had *significant availability impact*
Summary

- Start date / time of incident: 2015-10-06 00:30:00 PST
- Total duration of incident: 4 hours 5 minutes
- Postmortem Tracking bug: b24732745
  - Postmortem Owner: 
  - Contributors: 
- User impact:
OUTAGE ENDS

Root Cause(s)

Lessons Learned.

Action Items

Appendix - Other relevant tidbits
Where do failures happen?

No one network or plane dominates
How long do the failures last?

Shorter failures on B2

Durations much longer than outage budgets
What role does network evolution play?

70% of failures happen when a management operation is in progress.
The Zero Touch Network

Evolution is inevitable: Design for it!

{reliability, efficiency, scale} are NOT tradeoffs if network operation is fully intent driven.

Intent-driven Operation

Reliability, efficiency, scale
The Zero Touch Network

- All network operations are automated, requiring no operator steps beyond the instantiation of intent
- Changes applied to individual network elements are fully declarative, vendor-neutral, and derived by the network infrastructure from the high-level network-wide intent
- Any network changes are automatically halted and rolled-back if the network displays unintended behavior
- The infrastructure does not allow operations which violate network policies
The Zero Touch Network

- All network operations are automated, requiring no operator steps beyond the instantiation of intent.
- Changes applied to individual network elements are fully declarative, vendor-neutral, and derived by the network infrastructure from the high-level network-wide intent.
- Any network changes are automatically halted and rolled-back if the network displays unintended behavior.
- The infrastructure does not allow operations which violate network policies.
The Zero Touch Network

- All network operations are automated, requiring no operator steps beyond the instantiation of intent
- Changes applied to individual network elements are fully declarative, vendor-neutral and derived by the network infrastructure from the high-level network-wide intent
- Any network changes are automatically halted and rolled-back if the network displays unintended behavior
- The infrastructure does not allow operations which violate network policies
The Zero Touch Network

● All network operations are automated, requiring no operator steps beyond the instantiation of intent
● Changes applied to individual network elements are fully declarative, vendor-neutral and derived by the network infrastructure from the high-level network-wide intent
● Any network changes are automatically halted and rolled-back if the network displays unintended behavior
● The infrastructure does not allow operations which violate network policies
ZTN Architecture

operators -> Workflow Engine

“drain a link” -> Workflow API

Workflow API -> Update Network model

Topology -> Config

Network Management Layer

configuration, commands, telemetry

Network devices/ systems
Workflow Engine

- The workflow engine executes a goal-seeking workflow graph
- Workflows are expressed in a meta-language
- All interesting metrics of execution logged
- Workflows have the same test coverage as any software system
Network intent

- The workflow engine interacts with the intent-based network management infrastructure over transactional APIs.
- Workflow intents are expressed at the network-level, as changes to:
  - Topology
  - Config
  - Functional calls
Network Models

- **OpenConfig** ([www.openconfig.net](http://www.openconfig.net)) for vendor-neutral **configuration** model
  - YANG for data modeling, gRPC as transport
  - Both configuration and op-state models
  - BGP, MPLS, ISIS, L2, Optical-transport, ACL, policy...

- **“Unified Network Model”** for **topology**
  - Protocol Buffer based Google internal schema
  - Describes all layer-0/1/2/3 abstractions
Network Management Services

- Compose full config (vendor-neutral and vendor-specific) from topology/config intent update
- Provides secure transport of full config to network elements (OpenConfig+gRPC)
- Enforce Operational Policies
  - Rate limiting
  - Blast radius containment
  - Minimum survivable topology
Streaming Telemetry

network state changes observed by analyzing comprehensive time-series data stream

- Common schema for operational state data in OpenConfig
- stream data continuously -- with incremental updates
- Efficient, secure transport protocol, gRPC
Workflow Safety

- Ability to automatically check the safety of operations
- Ability to repeatedly validate the network state against the stated intent
- Ability to recognize “bad” network behavior
- Ability to roll back to the original state
Lessons learned from a decade of high-availability network design

Do not treat a change to the network as an exceptional event
Changes are common
Changes are **common**

↓

Make it safe to evolve the network **daily**
Changes are **common**

↓

Make it safe to evolve the network **daily**

↓

**Scale** just-in-time, scale often
Changes are **common**

↓

Make it safe to evolve the network **daily**

↓

**Scale** just-in-time, scale often

↓

Evolve into a **Zero Touch Network**
References

- B4: Experience With a Globally Deployed Software Defined WAN [sigcomm 2013]
- Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google’s Datacenter Network [Sigcomm 2015]
- Evolve or Die - High-Availability Design Principles Drawn from Google’s Network Infrastructure [sigcomm 2016]
- Andromeda: Google’s cloud networking stack
- OpenConfig: http://www.openconfig.net
- gRPC: http://www.grpc.io