Orion: Google's Software-Defined Networking Control Plane

Speaker: Chip Killian

Authors: Andrew D. Ferguson, Steve Gribble, Chi-Yao Hong, Charles Killian, Waqar Mohsin, Henrik Muehe, Joon Ong, Leon Poutievski, Arjun Singh, Lorenzo Vicisano, Richard Alimi, Shawn Shuoshuo Chen, Mike Conley, Subhasree Mandal, Karthik Nagaraj, Kondapa Naidu Bollineni, Amr Sabaa, Shidong Zhang, Min Zhu, and Amin Vahdat, Google

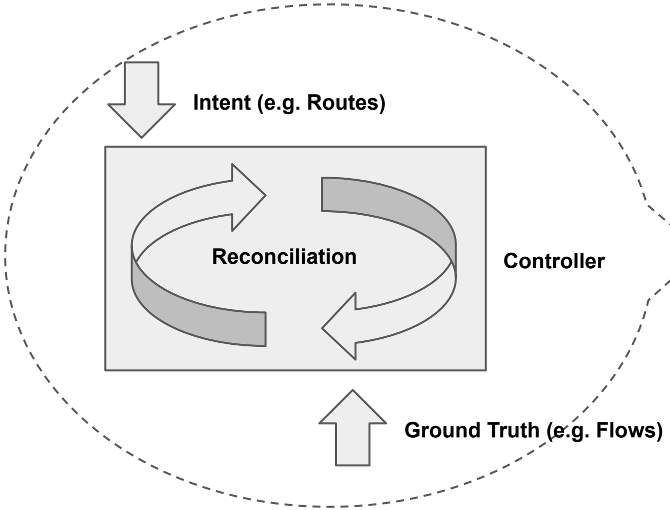
What is Orion?

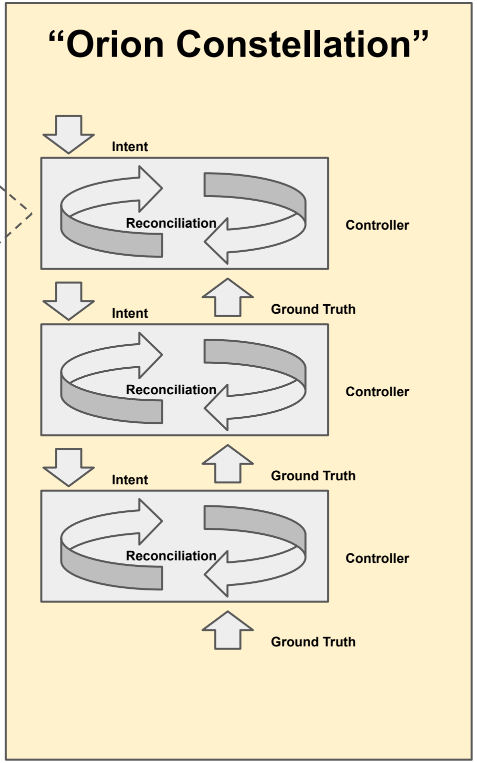
Orion is…

- Google’s 2nd generation SDN control plane
- Responsible for configuration, management, real-time network control
- In all our data center (Jupiter), campus, and private Wide Area (B4) networks
- In production for over 4 years
- A micro-services based architecture with a PubSub Network Information Base
- An evolving platform
    - biweekly releases, over 30 new significant capabilities, improved scale by 16x, improved availability by 50x (Jupiter) / 100x (B4), and improved convergence time by a factor of 40x

Software Design Principles

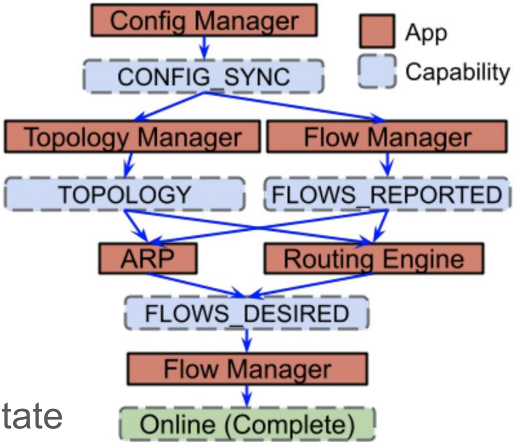
- Intent flows down
- Ground truth flows up
- Controller algorithm
  - Continually reconcile intent with ground truth
  - In the least disruptive way possible
- Controllers layer
  - One controller’s intent is another controller’s ground truth
    - Inter block routing → intra block routing → per-node flow programming
- Intent and ground truth are shared in a pub-sub Network Information Base (NIB)





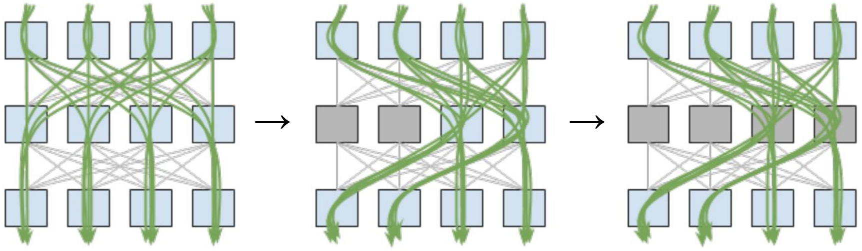
Orion: Continuous Reconciliation

- Ordinary reconciliation is continuous
  - E.g Network failures trigger rerouting
- Typical controller failure: Shared data preserved
  - Rebuild internal controller state as needed and continue
- Special case: Shared data lost
  - Capability Reconciliation: orderly reconciliation of lost data
  - Controllers:
    - Block on abstract capabilities reflecting required input state
    - Explicitly mark provided capabilities ready
  - Avoids need for strong durability of shared data
- Simplify: map all errors to a kind of controller failure
- For all reconciliations, convergence speed matters
  - Implies a loss of control or a period of lost traffic



Handling Dataplane ↔ Controller Disconnects

- Controllers manage switches over an asynchronous network
- Controller observed disconnect ≠ dataplane failure
- Experience: *Controller connection failures* are <u>more common</u> than dataplane network failures.
- Strategy:
  - Aggressively route around small uncorrelated failures (fail-closed)
  - Preserve current state in larger or correlated failures (fail-static)



(no-failure) (fail-closed) (fail-static)

Production Experience: Blast Radius Alignment

- Significant outage in 2019
- ⇧ Controller jobs were physically co-located near devices to limit blast radius!
- ⇩ Their virtual job management grouping was shared.
- A facility maintenance event
    - → a series of misconfigured behaviors
    - → disabled all Orion jobs in one virtual group
    - → failure larger than networks at that time were designed to tolerate
    - → a gap in our fail-static implementation for BGP caused routes to be withdrawn
    - → a user-visible outage
    - Even though data plane forwarding was generally not disrupted!

Conclusion: all management activities should be scoped and rate-limited

Thank You!

- Thank you to everyone who watches this!
- Thank you to my awesome coauthors and all the many collaborators who are not named on the paper!
- Questions? Comments? Want to join us?

orion-nsdi2021@google.com

Andrew D. Ferguson, Steve Gribble, Chi-Yao Hong, Charles Killian, Waqar Mohsin, Henrik Muehe, Joon Ong, Leon Poutievski, Arjun Singh, Lorenzo Vicisano, Richard Alimi, Shawn Shuoshuo Chen, Mike Conley, Subhasree Mandal, Karthik Nagaraj, Kondapa Naidu Bollineni, Amr Sabaa, Shidong Zhang, Min Zhu, and Amin Vahdat, *Google*