A Decentralized SDN Architecture for the WAN

Alexander Krentsel[1,2], Nitika Saran[3], Bikash Koley[1], Subhasree Mandal[1], Ashok Narayanan[1], Sylvia Ratnasamy[1,2], Ali Al-Shabibi[1], Anees Shaikh[1], Rob Shakir[1], Ankit Singla[1], Hakim Weatherspoon[3]

[1]Google, [2]UC Berkeley, [3]Cornell

Contact: akrentsel@berkeley.edu

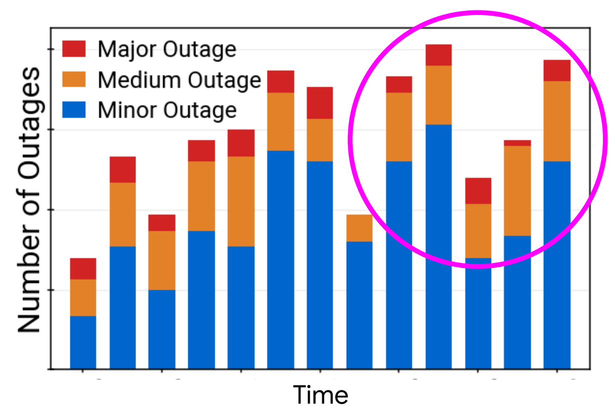
SIGCOMM 2024 SYDNEY

Paper QR code

Introduction: Outages Continue

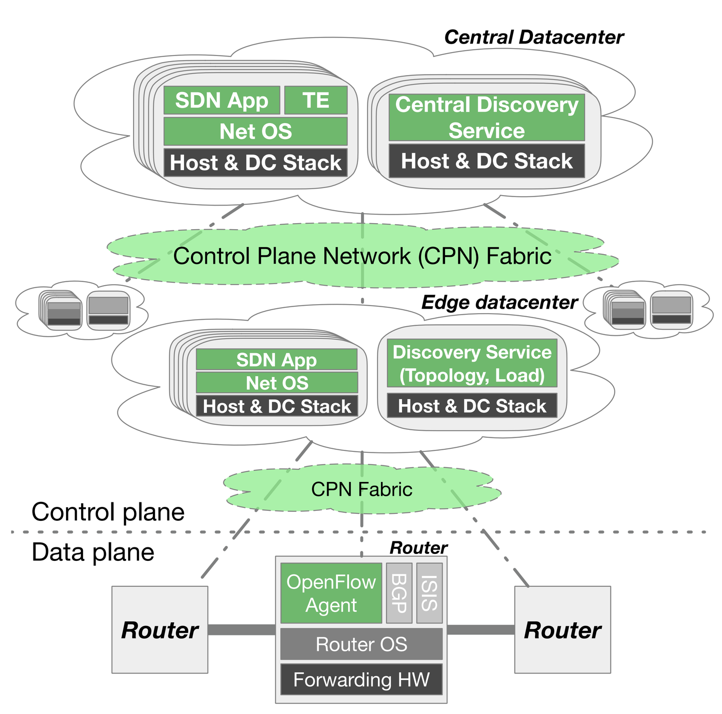
Global WANs continue to see large outages
- ✦ Global-scale WANs underpin planet-scale computing
- ✦ Outages continue *despite* decades of experience
- ✦ Small outages to be expected, large ones are unexpected
  - ...often complex, cascading root causes

What can we do at *design* time to limit the occurrence of complex failures?



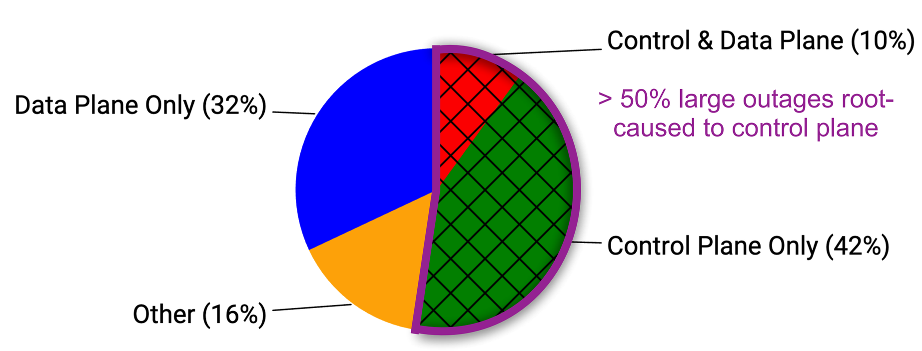
SDN WAN Architectures Today: Complexity Abounds

Running SDN for large-scale WANs requires much infrastructure:
- ✦ Hierarchy of control programs to provide aggregation
- ✦ Dedicated HW & SW to run them
- ✦ Control Plane Network to reach routers
- ✦ Discovery services to collect inputs
- ✦ Traditional protocols for fallback in case of split-brain



Traditional SDN Infrastructure

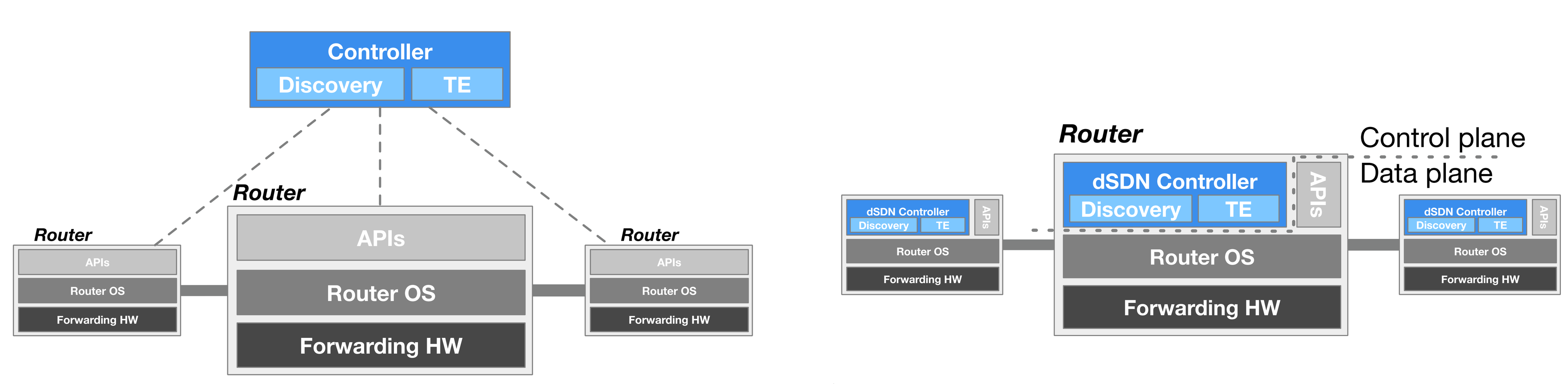
Majority of our SDN WAN large outages caused by control plane:



Approach: Decentralizing SDN

Key idea: decentralize SDN by *replicating* operator-written controller code on each router. Program paths via strict source routing to maintain "consensus-free" path selection.

Concretely, every router runs a dSDN controller that...
1. floods its local demand and link state; learns global network view
2. locally computes *all* paths (using a traffic eng. algorithm)
3. "programs" ingress end-to-end paths as *source routes*

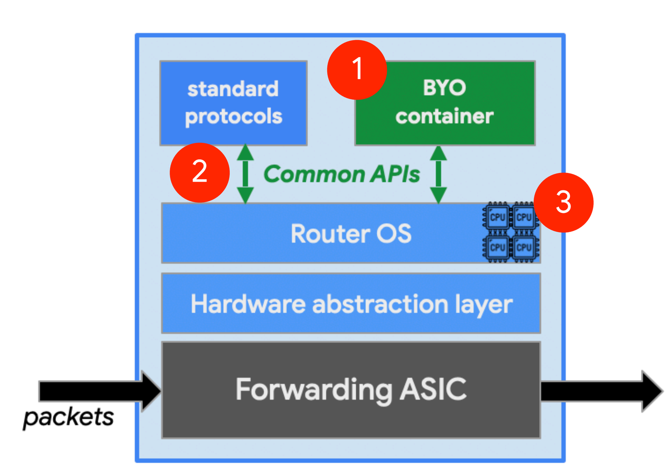


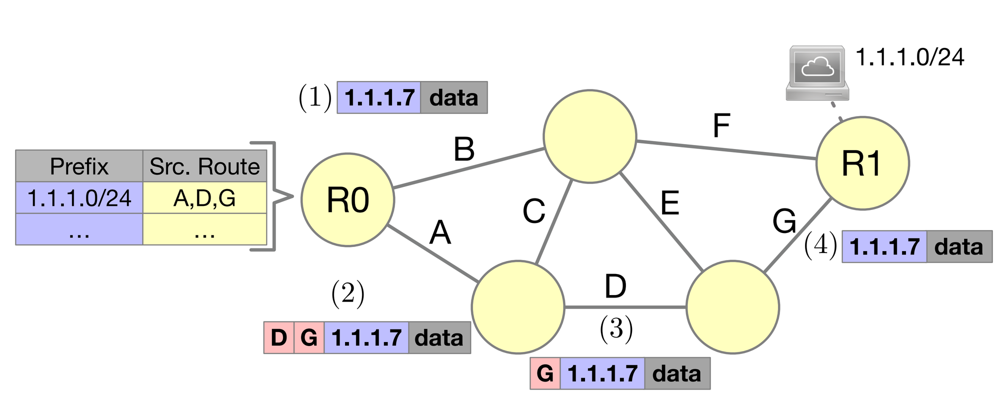
Centralized SDN Infrastructure → *Decentralized SDN Infrastructure*

Enabling Techniques: On-Box Containers, Source Routing

Router ecosystem developments provide new opportunities:
1. (1) Vendors support running 3rd-party containerized code on-box
   → operators can run custom control code on router CPU
2. (2) Standardized control APIs (gRIBI, gNMI, OF/P4) reach maturity
   → control code is uniform across vendors
3. (3) Expanded on-router CPU resources
   → from single-core to multi-core multi-GHz CPUs

Operator-defined control applications can run on-box





A packet's experience in a source-routed network

Source routing enables *simple* decentralization
- ✦ Maintains simplicity through *authority*: single point of responsibility per path
- ✦ Historically infeasible due to length of WAN paths

Enabled by hardware advancements and **novel encoding technique** that doubles information we can encode in headers

dSDN Achieves Benefits of SDN *and* Decentralization

Original SDN Benefits

- ✦ **Operator-defined code**, which enabled innovation
  - ✔ *running operator-written containers on the router*
- ✦ **Optimized** computations (TE) on global view of network
  - ✔ *new APIs + simple dissemination → global view*
- ✦ **Simplicity** of "consensus free" path selection
  - ✔ *source-routing; "ingress" router authoritatively decides path*

Decentralization Benefits

- ✦ Drastically **fewer** external **dependencies**
  - ✔ *control plane running entirely in-band*
- ✦ Distributed **survivability**
  - ✔ *no central point of failure*

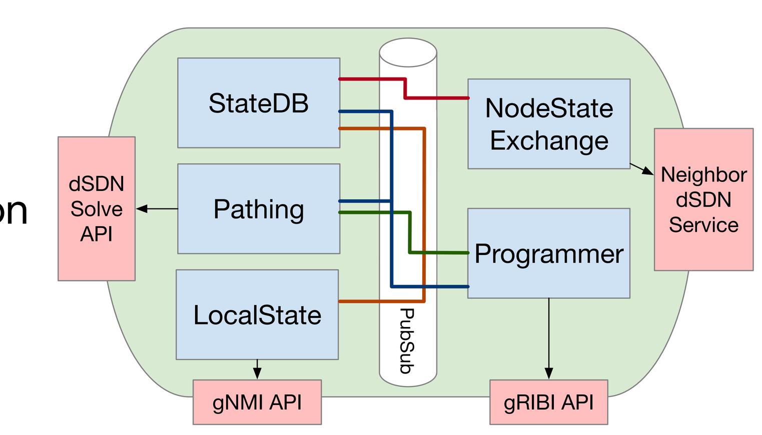
Evaluation: Simplicity *Improves* Performance

Primary goal: cutting complexity from the WAN architecture

Removed	Added
(1) Central controller jobs	On-router containers
(2) Regional controller jobs	
(3) Dedicated server hardware	
(4) Control plane network	
(5) Instrumentation services	
(6) Traditional protocols	

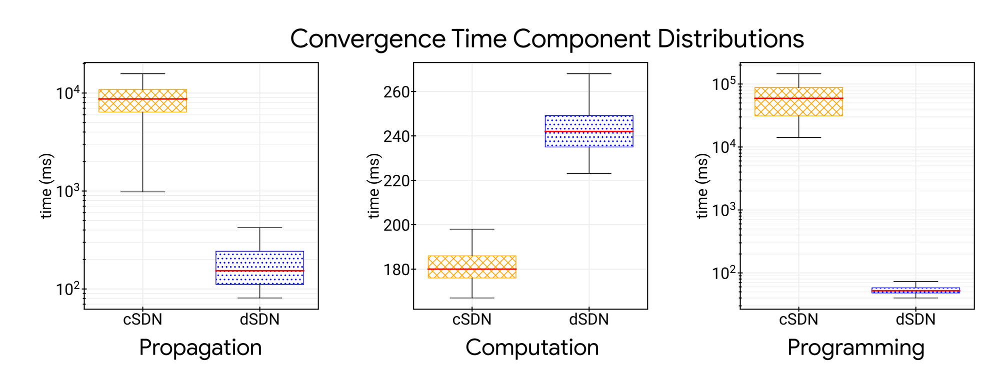
Success! Do we lose anything in the process?
On the contrary, performance *improves*...

Methodology
- ✦ Built production-grade dSDN Implementation
- ✦ Profiled traditional SDN performance in production
- ✦ Profiled dSDN on production hardware feeding in production B4 topology & demand
- ✦ Simulated historical failure events in high-fidelity simulator



dSDN Implementation Architecture

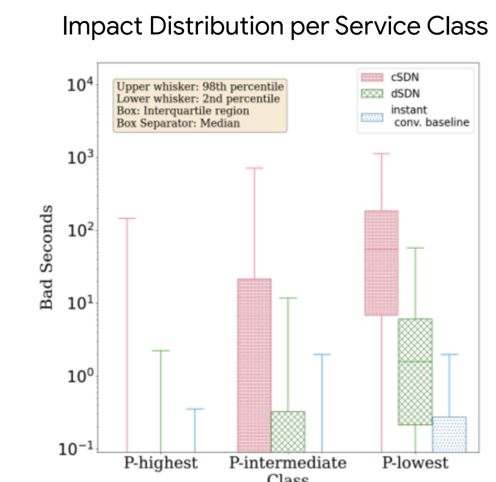
Convergence time: 120x-150x faster
- ✦ Propagation & programming time 100-1000x faster
- ✦ Computation ~30% slower due to weaker router compute



Convergence impact: 20x-60x faster
- ✦ "Impact" comprises both number of flows affected and amount of time
- ✦ ***Bad seconds*** metric defined in paper captures both

	cSDN	dSDN
Highest Priority	146.9	2.23
Lowest Priority	1122.9	57.3

Cumulative Impact (Bad Seconds) Across Flows at 98th Percentile



Conclusion: dSDN as a New Point in Design Space

We've spent...
- ➡ 30 years trying to make decentralized protocols work more efficiently with more features
- ➡ 15 years trying to make SDN-based networks work more reliability

We present **dSDN**, a new point in the design space enabled by new hardware capabilities...
- ➡ achieves the best of both by decentralizing the SDN controller in a way that...
  - ★ maintains benefits of SDN
  - ★ significantly simplifies the control plane infrastructure
  - ★ improves convergence performance