Challenges in Network Debugging

- Modern networks are large and complex
- Complex interactions between different devices increase chance of misconfiguration
  - Routers, firewalls, switches, NAT boxes, etc.
- Bugs in software implementation that only get triggered under certain conditions
  - May cause large black outs if not detected while testing the network
- Difficult to test the entire state space
  - \(2^{32}\) IP addresses, \(2^{16}\) port addresses, etc.

Our Approach: Data Plane Verification

- VeriFlow checks network-wide invariants in real time using data-plane state
- Absence of routing loops and black holes, access control violations, etc.
- It functions by
  - Monitoring dynamic changes in the network
  - Constructing a model of the network behavior
  - Using custom algorithms to automatically derive whether the network contains errors

VeriFlow Operation

- VeriFlow checks network-wide invariants in real time
  - Configuration
    - Closer to actual network behavior
  - Control plane
    - Less prediction
  - Data-plane state
    - Unified analysis for multiple control-plane protocols
  - Network behavior
    - Can catch control-plane implementation bugs

Microbenchmark Results

- 97.8% of the updates were verified within 1 millisecond
- Experiment with NOX and Mininet
  - TCP connection attempts (per sec)
    - The largest reduction in throughput was only 12.8%

Related Work

- Header space analysis: Static checking for networks, NSDI 2012
- A NICE way to test OpenFlow applications, NSDI 2012
- Abstractions for network update, SIGCOMM 2012
- Debugging the data plane with Anteater, SIGCOMM 2011
- Can the production network be the testbed?, OSDI 2010
- FlowChecker: Configuration analysis and verification of federated OpenFlow infrastructures, SafeConfig 2010
- Network configuration in a box: Towards end-to-end verification of network reachability and security, ICNP 2009