VeriFlow: Verifying Network-Wide Invariants in Real Time

Ahmed Khurshid, Wenxuan Zhou,
Matthew Caesar, P. Brighten Godfrey
University of Illinois at Urbana-Champaign
Modern networks are complex.
Modern networks are complex.
Modern networks are complex.
Modern networks are complex.

E.g., Loops, Black holes, Security Violations, ...

Wednesday, August 15, 2012
Modern networks are complex.

E.g., Loops, Black holes, Security Violations, ...
Motivation

Modern networks are complex.

Serious consequences!

E.g., Loops, Black holes, Security Violations, ...

Wednesday, August 15, 2012
Modern networks are complex.

Serious consequences!

E.g., Loops, Black holes, Security Violations, ...

Debugging the data plane

- *Diagnose problems as close as possible to actual network behavior*
- *Data plane is a “narrower waist” than configuration*
Motivation

What if we can detect bugs in real time? (~1 ms)

• Provide immediate warning
• Block dangerous changes
Motivation

What if we can detect bugs in real time? (~1 ms)

- Provide immediate warning
- Block dangerous changes

Is it possible to check network-wide invariants in real time as the network evolves?
Challenge 1: Obtaining real time view of network

- Solution: interpose between SDN controller and devices
**Challenge 1: Obtaining real time view of network**

- **Solution**: interpose between SDN controller and devices

**Challenge 2: Verification speed**

- **Solution**: Formal methods?
Motivation

Challenge 1: Obtaining real time view of network

- Solution: interpose between SDN controller and devices

Challenge 2: Verification speed

- Solution: Formal methods? No, too slow!

Anteater, Mai, Khurshid, Agarwal, Caesar, Godfrey, and King. (SIGCOMM 11)
ConfigChecker, Al-Shaer, Marrero, El-Atawy, and ElBadawi. (ICNP 09)
HSA, Kazemian, Varghese, and McKeown. (NSDI 12)
Outline

• Motivation
• Design
• Evaluation
• Conclusion
Our Approach: VeriFlow
Our Approach: VeriFlow

SDN Controller

Monitor all updates!

VeriFlow
Our Approach: VeriFlow

Monitor all updates!
Our Approach: VeriFlow

Monitor all updates!
Overview

VeriFlow

Generate Equivalence Classes

Updates
Overview

Updates

Generate Equivalence Classes

VeriFlow
Overview

VeriFlow

Generate Equivalence Classes

Generate Forwarding Graphs

Updates

Wednesday, August 15, 2012
Overview

VeriFlow

Generate Equivalence Classes

Generate Forwarding Graphs

Updates
Overview

VeriFlow

- Generate Equivalence Classes
- Generate Forwarding Graphs
- Run Queries

Updates
1. Limit the Search Space

VeriFlow

Updates
1. Limit the Search Space

VeriFlow

Updates

Generate Equivalence Classes
1. Limit the Search Space

VeriFlow

Generate Equivalence Classes

Updates

*Equivalence class:* Packets experiencing the same forwarding actions throughout the network.
1. Limit the Search Space

**VeriFlow**

**Generate Equivalence Classes**

*Equivalence class:* Packets experiencing the same forwarding actions throughout the network.

Fwd’ing rules

Wednesday, August 15, 2012
1. Limit the Search Space

VeriFlow

Generate Equivalence Classes

Equivalence class: Packets experiencing the same forwarding actions throughout the network.

Fwd’ing rules 0.0.0.0/1

Updates
I. Limit the Search Space

**VeriFlow**

**Generate Equivalence Classes**

**Equivalent class:** Packets experiencing the same forwarding actions throughout the network.

**Fwd’ing rules**

0.0.0.0/1

64.0.0.0/3

**Updates**
1. Limit the Search Space

VeriFlow

Generate Equivalence Classes

Equivalence class: Packets experiencing the same forwarding actions throughout the network.

Fwd’ing rules

Equiv classes

Updates

0.0.0.0/1

64.0.0.0/3
1. Limit the Search Space

VeriFlow

Generate Equivalence Classes

Updates

Equivalence class: Packets experiencing the same forwarding actions throughout the network.

Find only equivalence classes affected by the update using a trie-based data structure

Fwd’ing rules
Equiv classes

0.0.0.0/1 64.0.0.0/3
2. Represent Forwarding Behavior

VeriFlow

Generate Equivalence Classes

Updates
2. Represent Forwarding Behavior

VeriFlow

Updates

Generate Equivalence Classes
2. Represent Forwarding Behavior

VeriFlow

Generate Equivalence Classes

Generate Forwarding Graphs

Updates
2. Represent Forwarding Behavior

VeriFlow

Generate Equivalence Classes

Generate Forwarding Graphs

Updates

Forwarding graphs:
2. Represent Forwarding Behavior

VeriFlow

Updates

Generate Equivalence Classes

Generate Forwarding Graphs

Forwarding graphs:

All the info to answer queries!
3. Run Graph Alg. to Check Invariants

VeriFlow

Generate Equivalence Classes

Generate Forwarding Graphs

Updates

Wednesday, August 15, 2012
3. Run Graph Alg. to Check Invariants

VeriFlow

Generate Equivalence Classes

Generate Forwarding Graphs

Updates
3. Run Graph Alg. to Check Invariants

VeriFlow

Generate Equivalence Classes → Generate Forwarding Graphs → Run Queries

Updates
3. Run Graph Alg. to Check Invariants

VeriFlow

Generate Equivalence Classes → Generate Forwarding Graphs → Run Queries

Updates

Reachability Queries:

- Black holes,
- Routing loops,
- Isolation of multiple VLANs,
- Access control policies,

... General Queries
3. Run Graph Alg. to Check Invariants

**VeriFlow**

- Generate Equivalence Classes
- Generate Forwarding Graphs
- Run Queries

**Updates**

Wednesday, August 15, 2012
3. Run Graph Alg. to Check Invariants

VeriFlow

Generate Equivalence Classes

Generate Forwarding Graphs

Run Queries

Updates
3. Run Graph Alg. to Check Invariants

- Generate Equivalence Classes
- Generate Forwarding Graphs
- Run Queries

VeriFlow

Updates

Good rules

Wednesday, August 15, 2012
3. Run Graph Alg. to Check Invariants

- **Updates**
  - Generate Equivalence Classes
  - Generate Forwarding Graphs
  - Run Queries

- **VeriFlow**
  - Good rules
  - Bad rules

- Network diagram with interconnected nodes representing systems or processes.
3. Run Graph Alg. to Check Invariants

Diagnosis report
- Type of invariant violation
- Affected set of packets

VeriFlow
- Generate Equivalence Classes
- Generate Forwarding Graphs
- Run Queries

Updates
- Good rules
- Bad rules
Outline

• Motivation
• Design
• Evaluation
• Conclusion
A new experiment not in the paper [with Kelvin Zou]

- Mininet OpenFlow network
- 172 switches, 172 hosts
- NOX controller, learning switch app
- TCP connections between random pairs of hosts
A new experiment not in the paper [with Kelvin Zou]
• Mininet OpenFlow network
• 172 switches, 172 hosts
• NOX controller, learning switch app
• TCP connections between random pairs of hosts
Microbenchmark Runtime

CDF

Graph cache update
Equivalence class search
Graph build
Query
Total verification

Microseconds

Microbenchmark Runtime

Wednesday, August 15, 2012
Microbenchmark Runtime

CDF

Graph cache update
Equivalence class search
Graph build
Query
Total verification

99% of updates verified within 200 µs
Throughput

Verification overhead 7% on average
Perform Verification with minimal overhead

Throughput

Verification overhead 7% on average

TCP connection setup latency (seconds)
VeriFlow achieves real-time verification

- A layer between SDN controller & network devices
- Rigorous checking within hundreds of μs
Conclusion

VeriFlow achieves real-time verification

- A layer between SDN controller & network devices
- Rigorous checking within hundreds of μs

Thank you.