

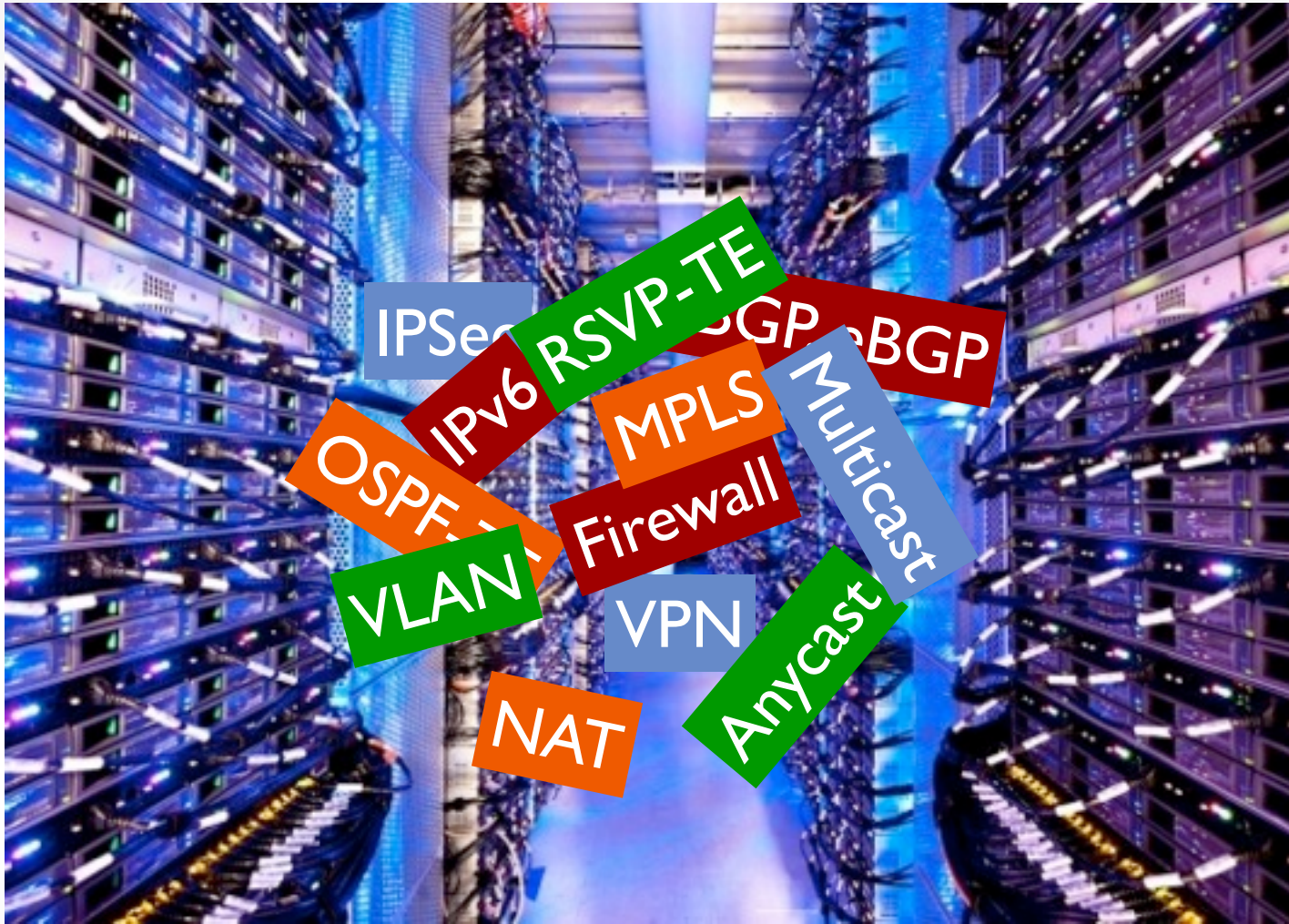
VeriFlow: Verifying Network-Wide Invariants in Real Time

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,

...



Modern networks are complex.



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

...



Modern networks are complex.



Serious consequences!



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

...



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*



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

- Provide immediate warning
- Block dangerous changes



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?

U6fW0Rk 6V0|V62j

U6fW0Rk 6V0|V62j

Motivation





Challenge I: 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?



Challenge 1: Obtaining real time view of network

- Solution: interpose between SDN controller and devices

Challenge 2: Verification speed

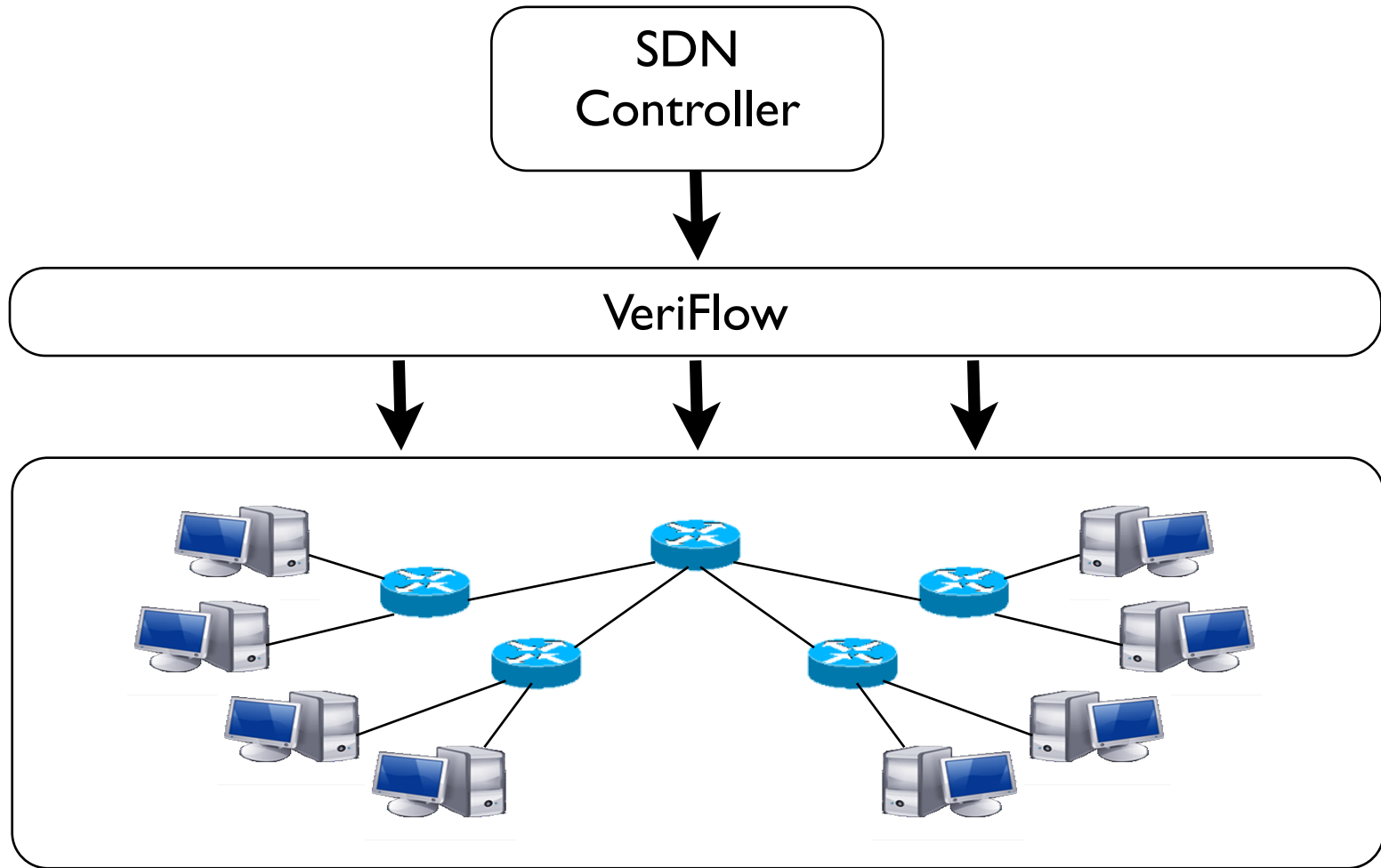
- Solution: Formal methods? *No, too slow!*

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)

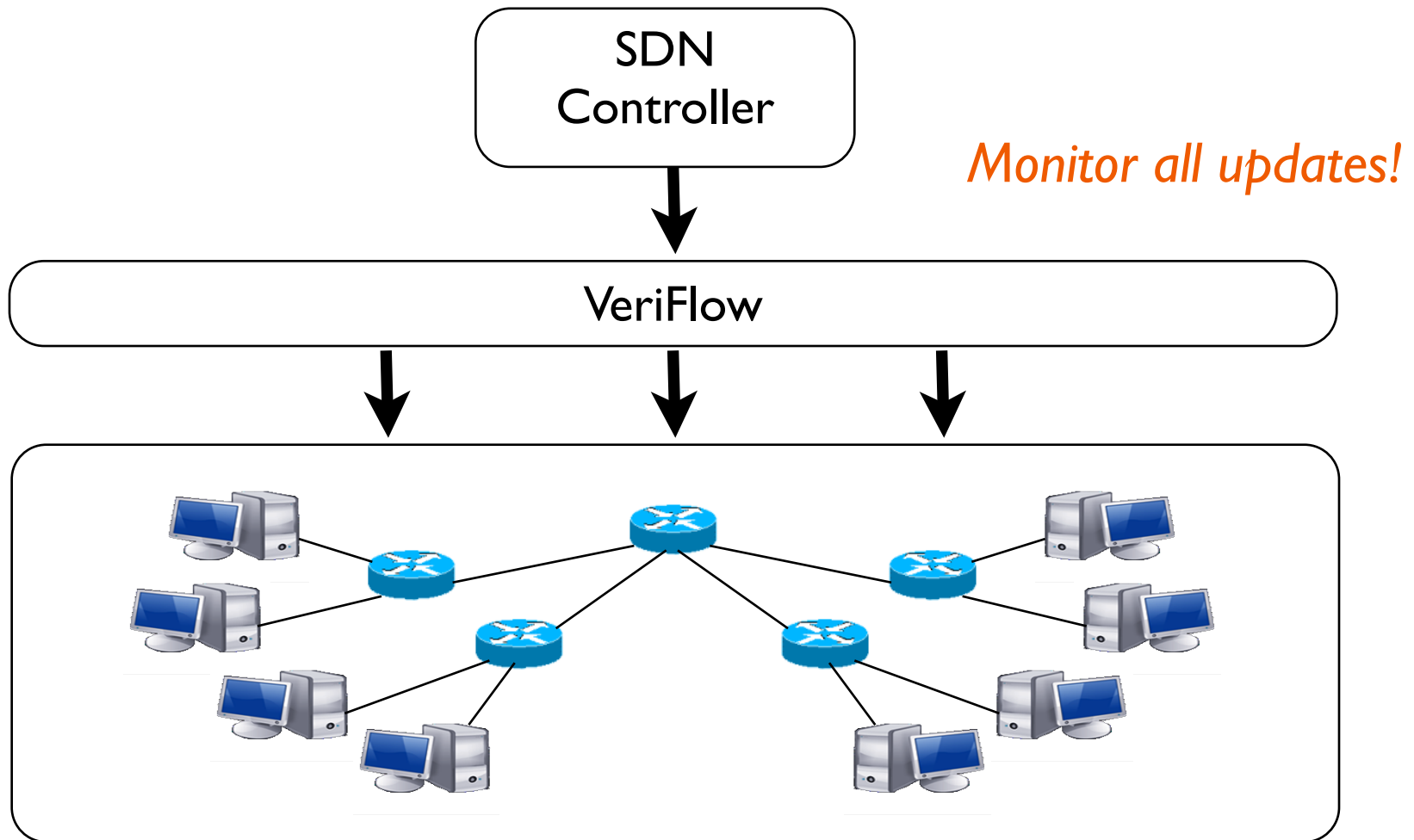


- *Motivation*
- Design
- Evaluation
- Conclusion

Our Approach: VeriFlow



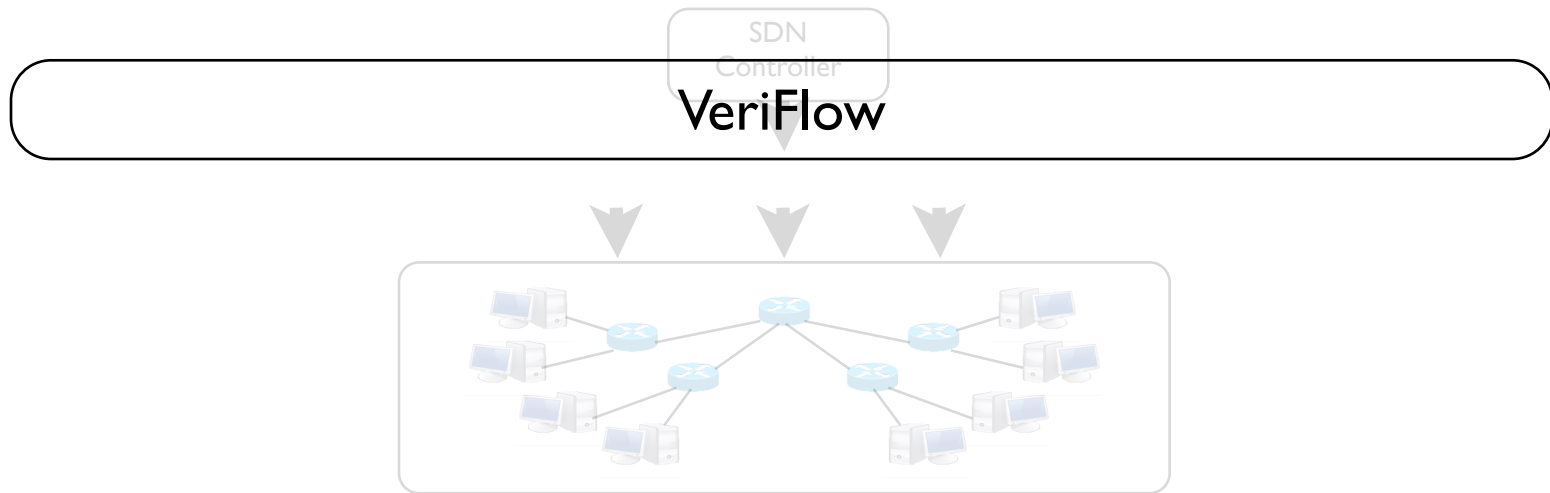
Our Approach: VeriFlow



Our Approach: VeriFlow



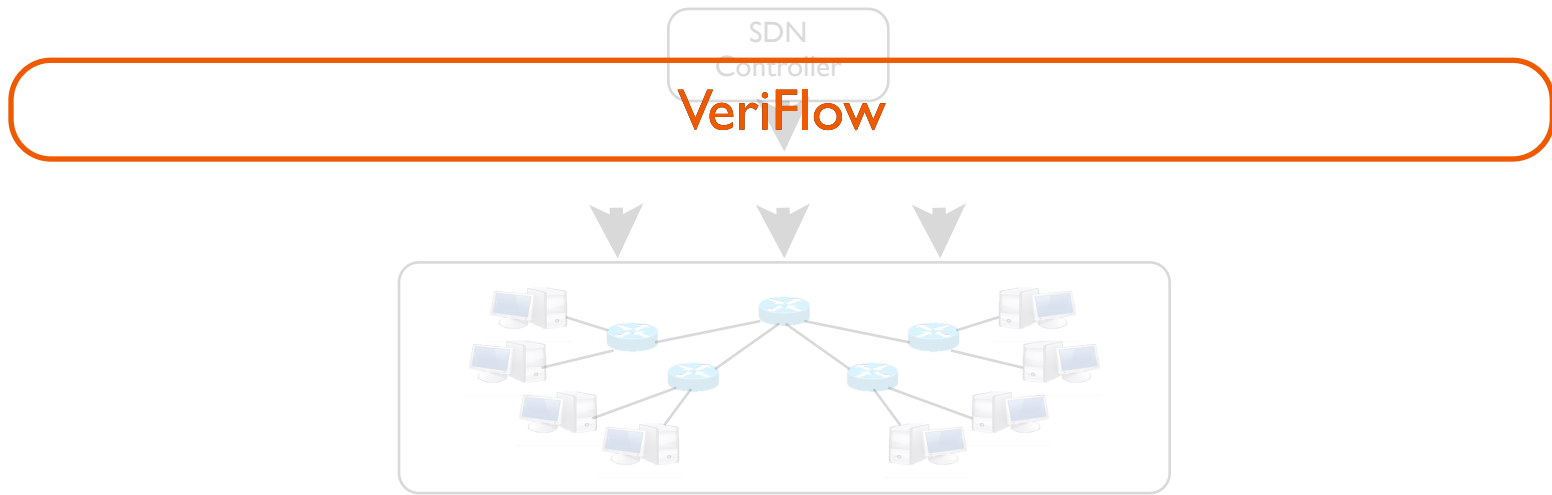
Monitor all updates!



Our Approach: VeriFlow



Monitor all updates!





VeriFlow

Updates





VeriFlow

Generate
Equivalence
Classes

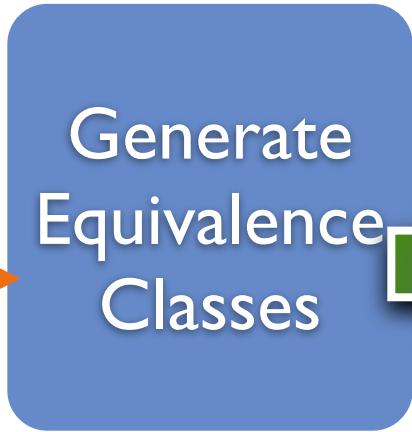
Updates





VeriFlow

Updates





VeriFlow

Generate
Equivalence
Classes



Generate
Forwarding
Graphs

Updates





VeriFlow

Generate
Equivalence
Classes



Generate
Forwarding
Graphs



Updates





VeriFlow

Updates



Generate
Equivalence
Classes



Generate
Forwarding
Graphs



Run
Queries

I. Limit the Search Space



VeriFlow

Updates



I. Limit the Search Space



VeriFlow

Generate
Equivalence
Classes

Updates



I. Limit the Search Space



VeriFlow

Generate
Equivalence
Classes

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

Updates



I. Limit the Search Space



VeriFlow

Generate
Equivalence
Classes

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

Fwd'ing rules



I. 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



I. Limit the Search Space



VeriFlow

Generate
Equivalence
Classes

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

Fwd'ing rules



I. Limit the Search Space

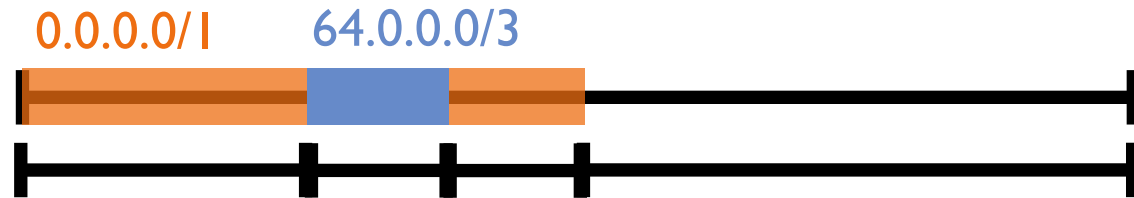


VeriFlow

Generate
Equivalence
Classes

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

Fwd'ing rules
Equiv classes



I. Limit the Search Space

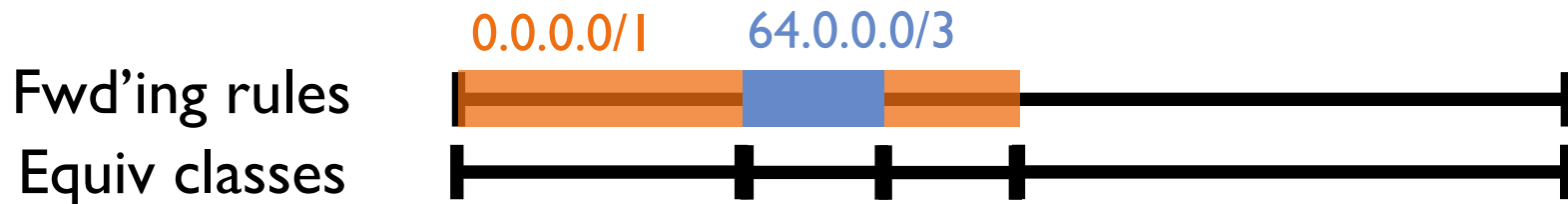


VeriFlow

Generate
Equivalence
Classes

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

Updates



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

2. Represent Forwarding Behavior



VeriFlow

Generate
Equivalence
Classes

Updates



2. Represent Forwarding Behavior



VeriFlow

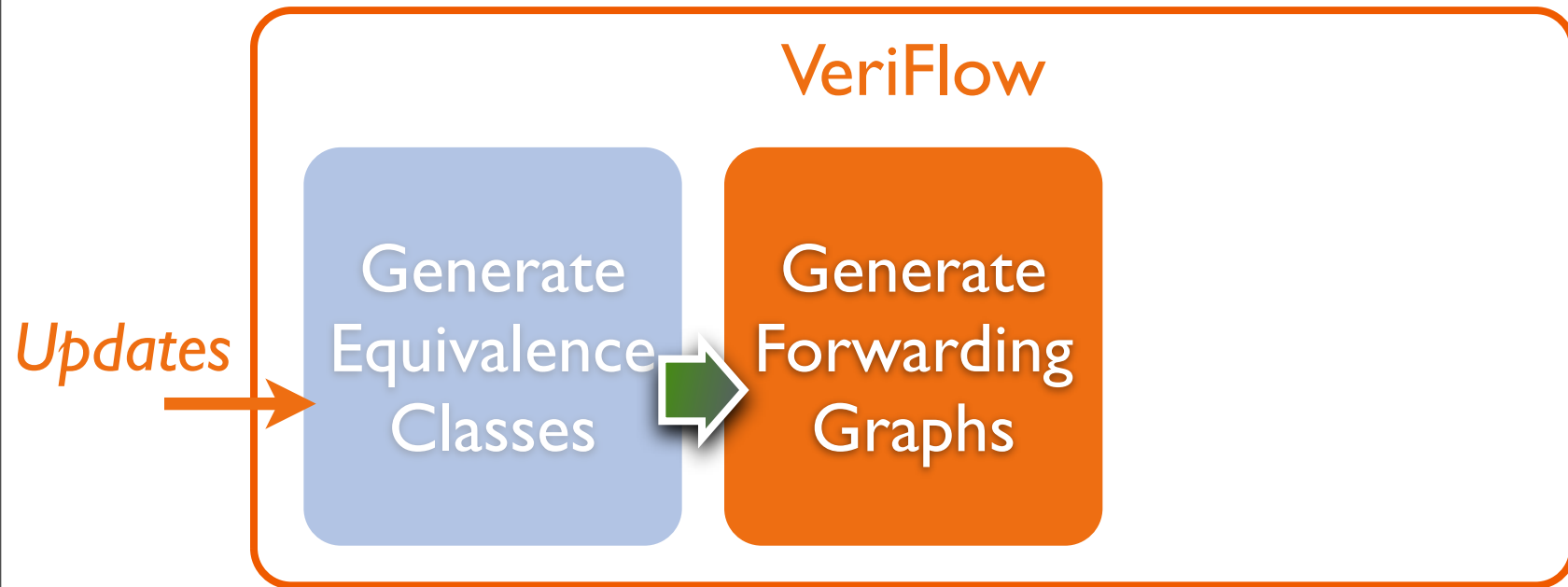
Generate
Equivalence
Classes



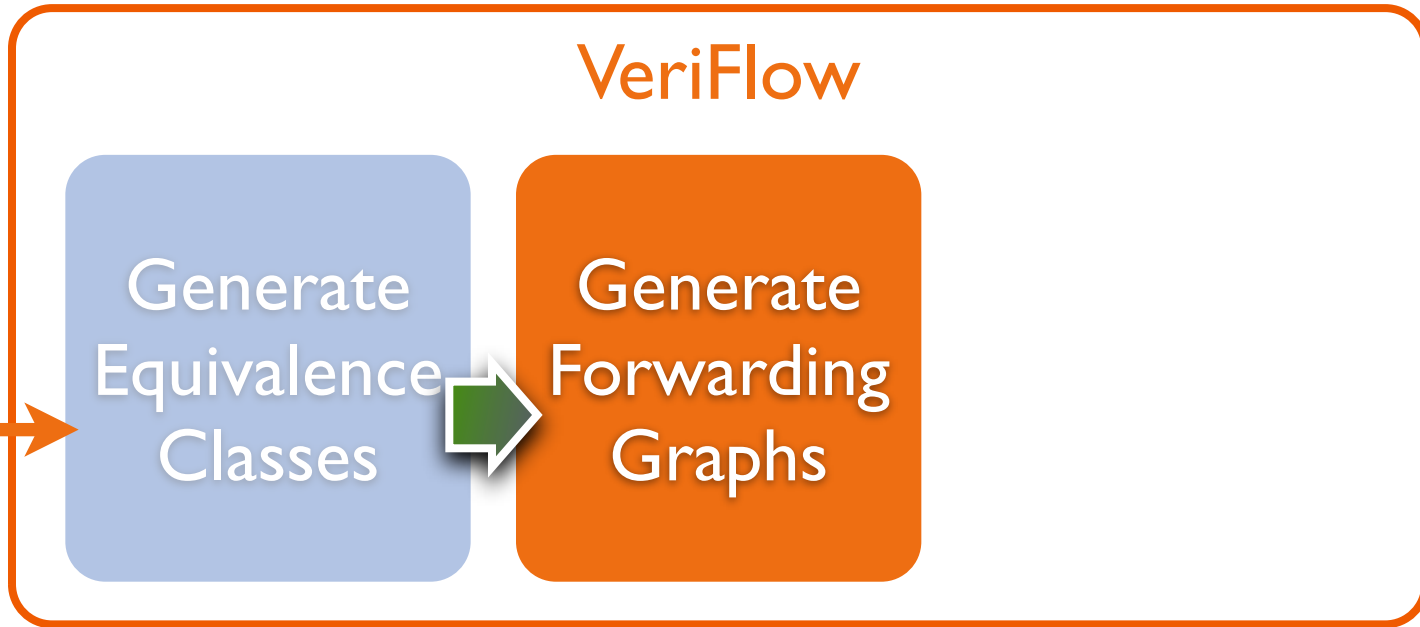
Updates



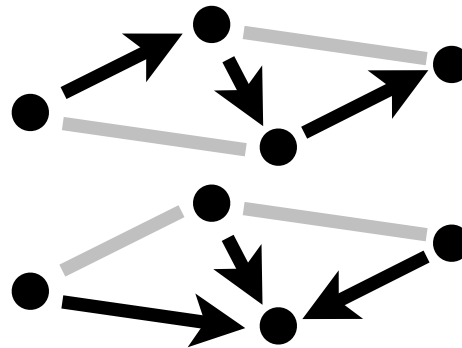
2. Represent Forwarding Behavior



2. Represent Forwarding Behavior

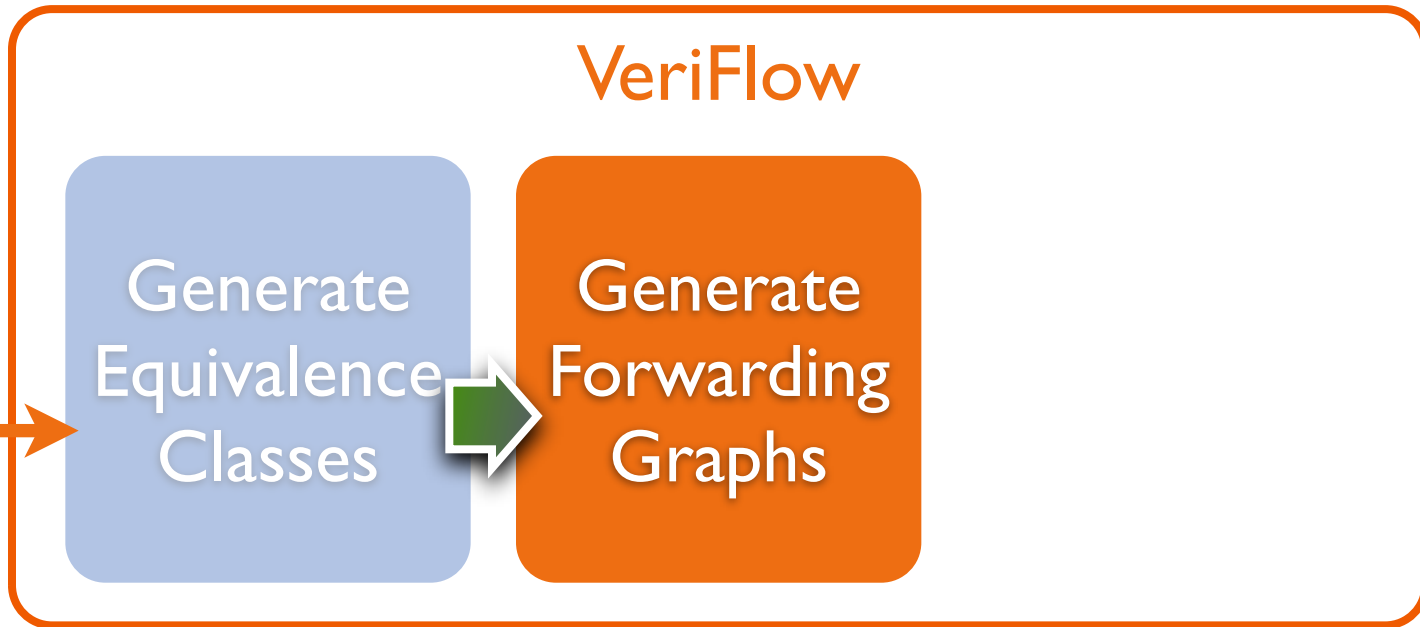


Forwarding graphs:

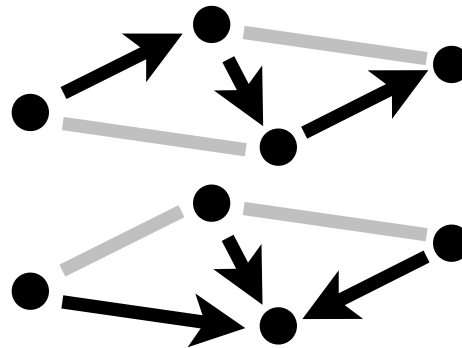




2. Represent Forwarding Behavior



Forwarding graphs:



All the info to answer queries!

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

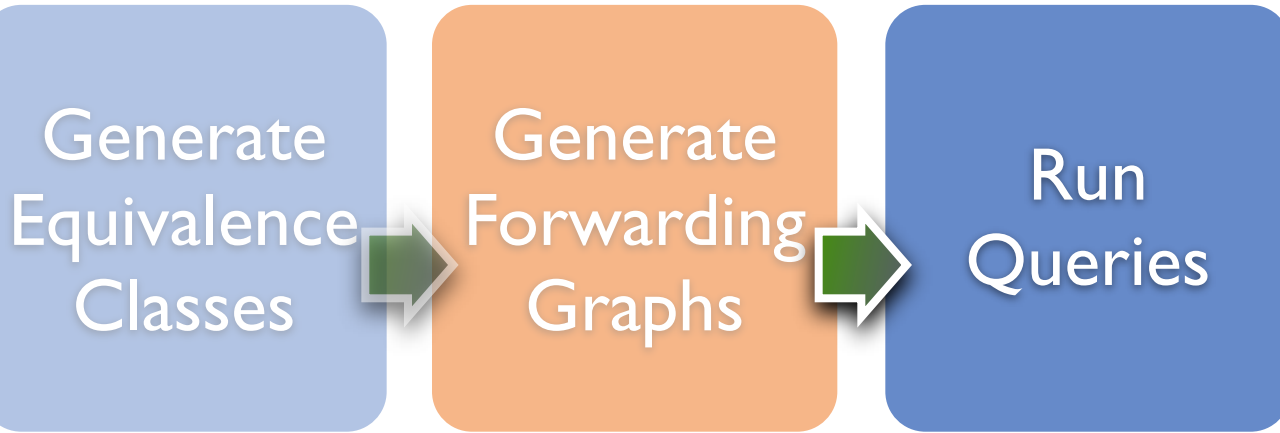
Updates



3. Run Graph Alg. to Check Invariants



VeriFlow



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,*

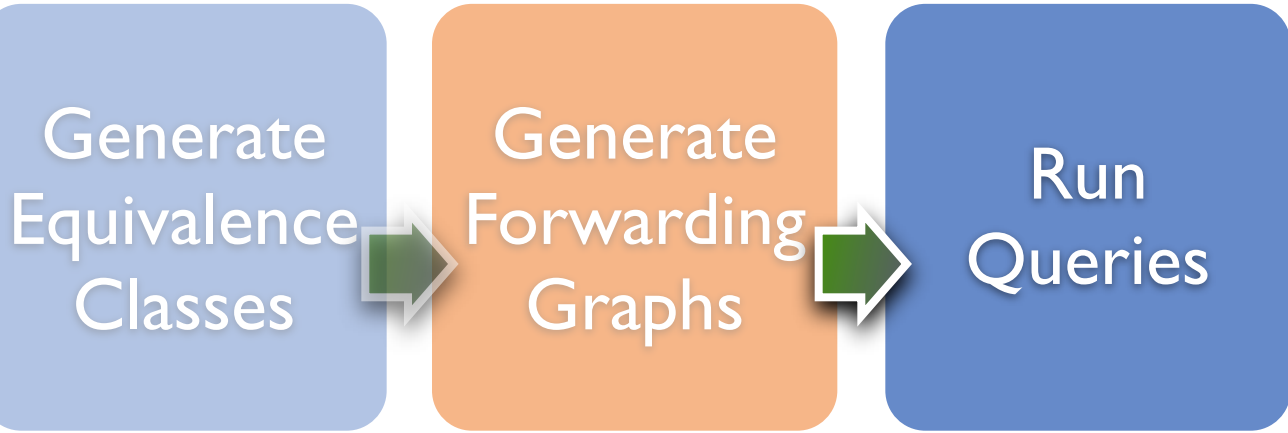
...

General Queries

3. Run Graph Alg. to Check Invariants



VeriFlow





3. Run Graph Alg. to Check Invariants

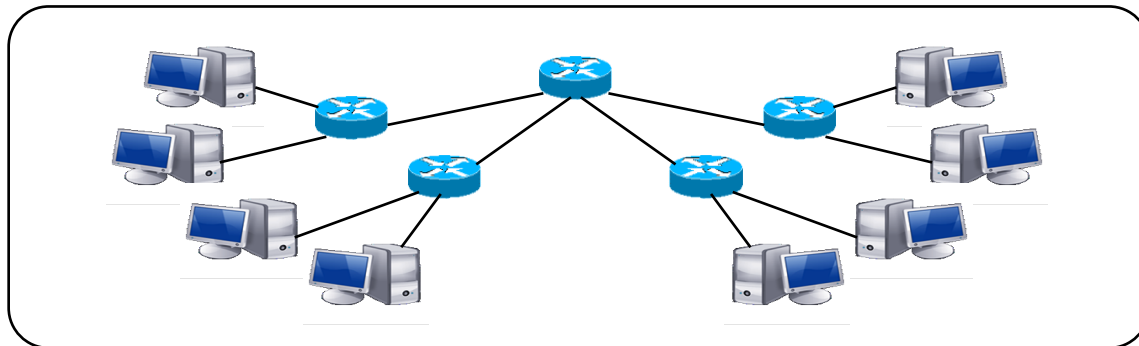
VeriFlow

Generate
Equivalence
Classes

Generate
Forwarding
Graphs

Run
Queries

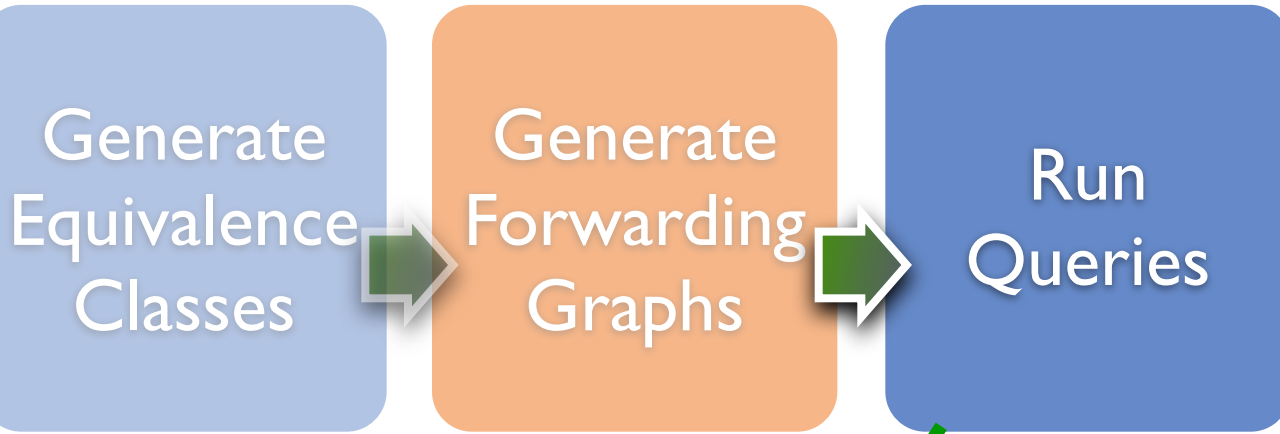
Updates





3. Run Graph Alg. to Check Invariants

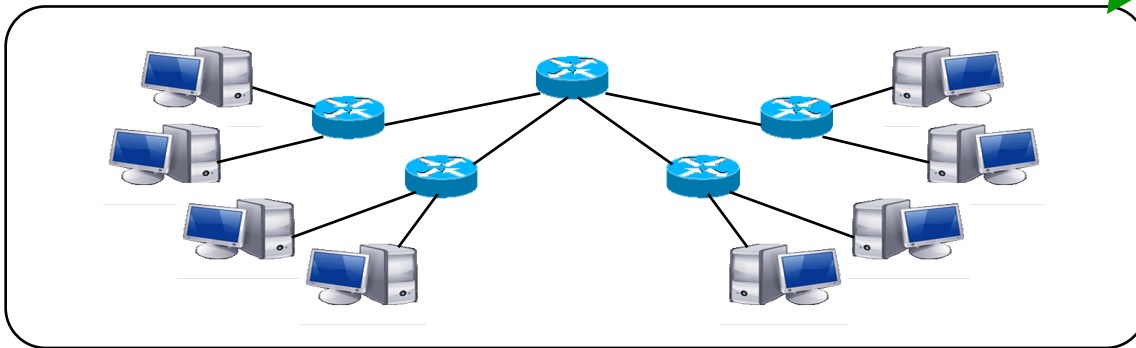
VeriFlow



Updates



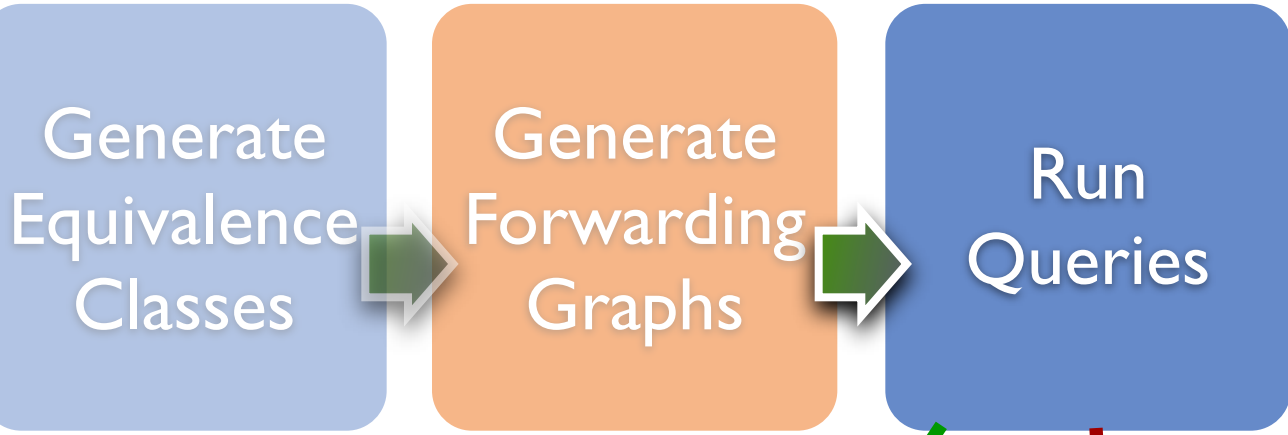
Good rules





3. Run Graph Alg. to Check Invariants

VeriFlow



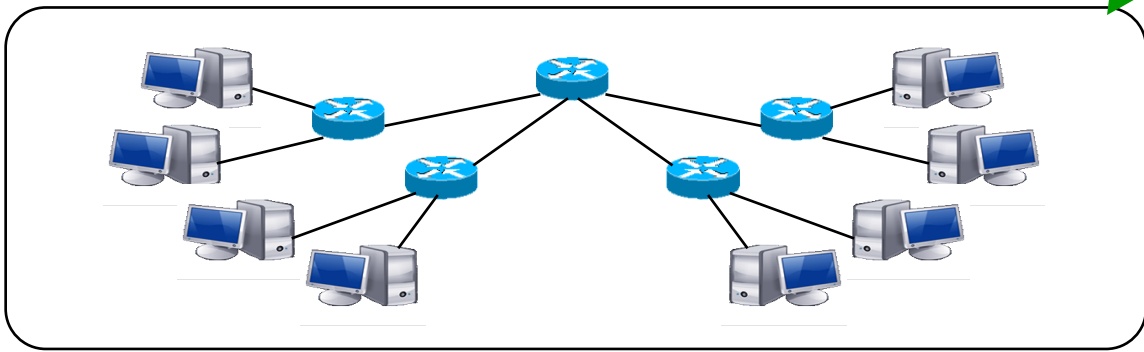
Updates



Good rules



Bad rules



3. Run Graph Alg. to Check Invariants



VeriFlow

Generate
Equivalence
Classes

Generate
Forwarding
Graphs

Run
Queries

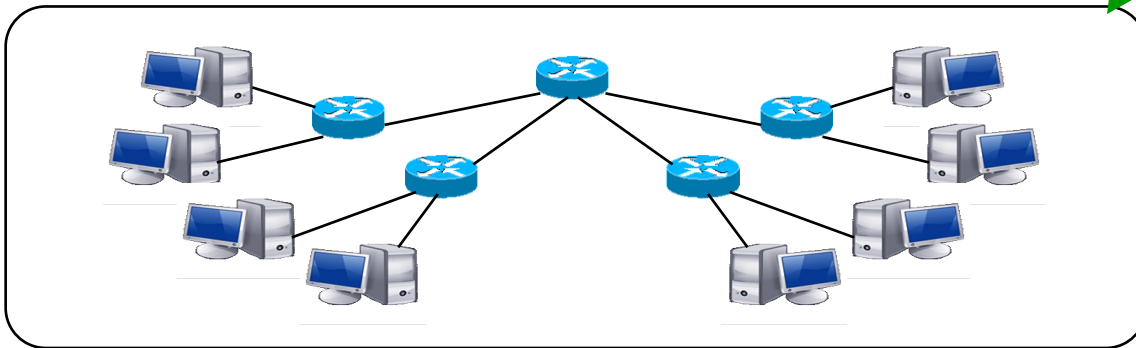
Updates

Good rules

Bad rules

Diagnosis report

- Type of invariant violation
- Affected set of packets





- *Motivation*
- *Design*
- **Evaluation**
- **Conclusion**

Evaluation Setup

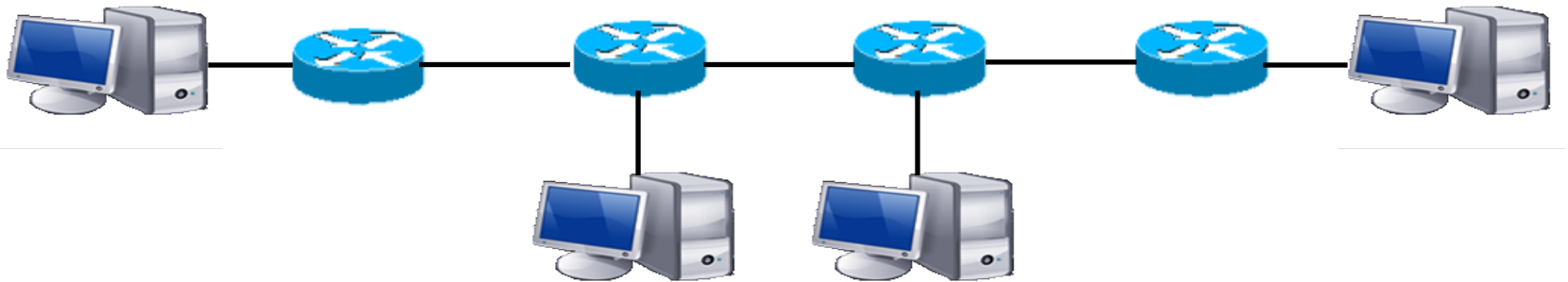


A new experiment not in the paper [with Kelvin Zou]

- Mininet OpenFlow network
- 172 switches, 172 hosts
- NOX controller, learning switch app
- TCP connections between random pairs of hosts

NOX Controller

VeriFlow

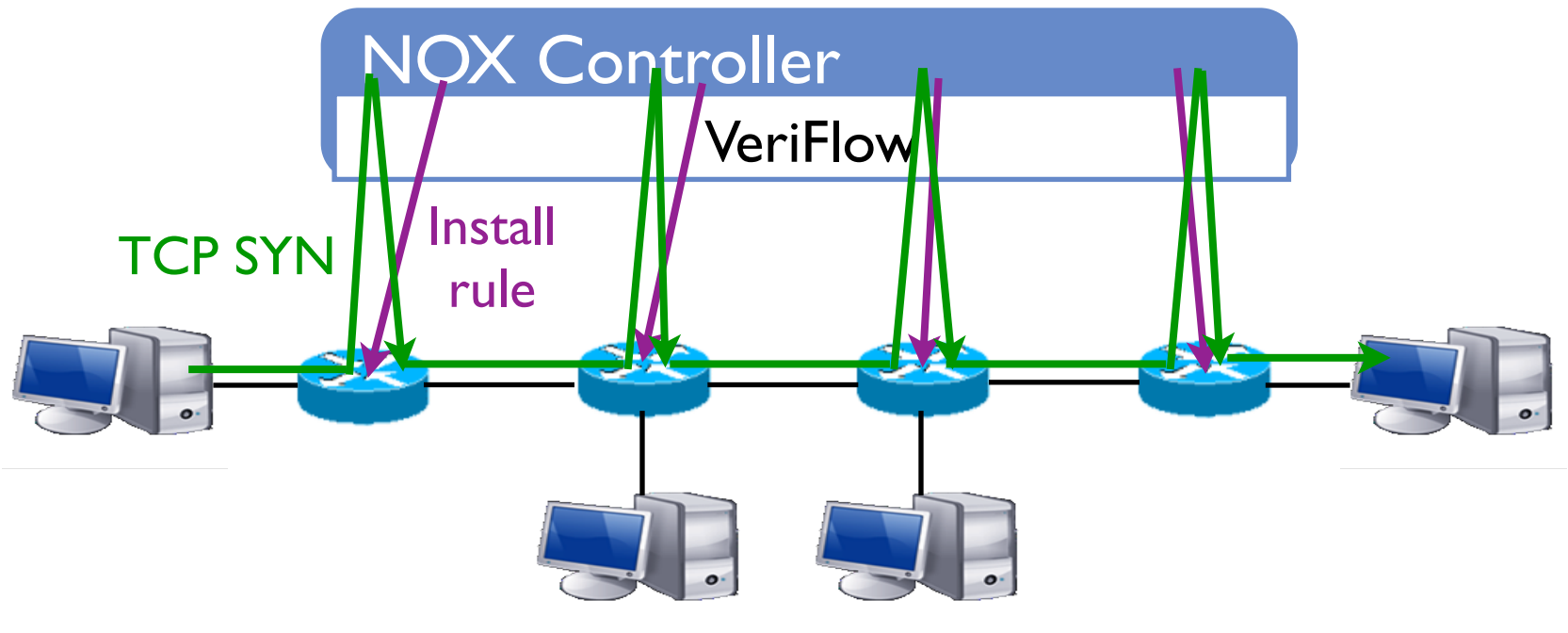


Evaluation Setup

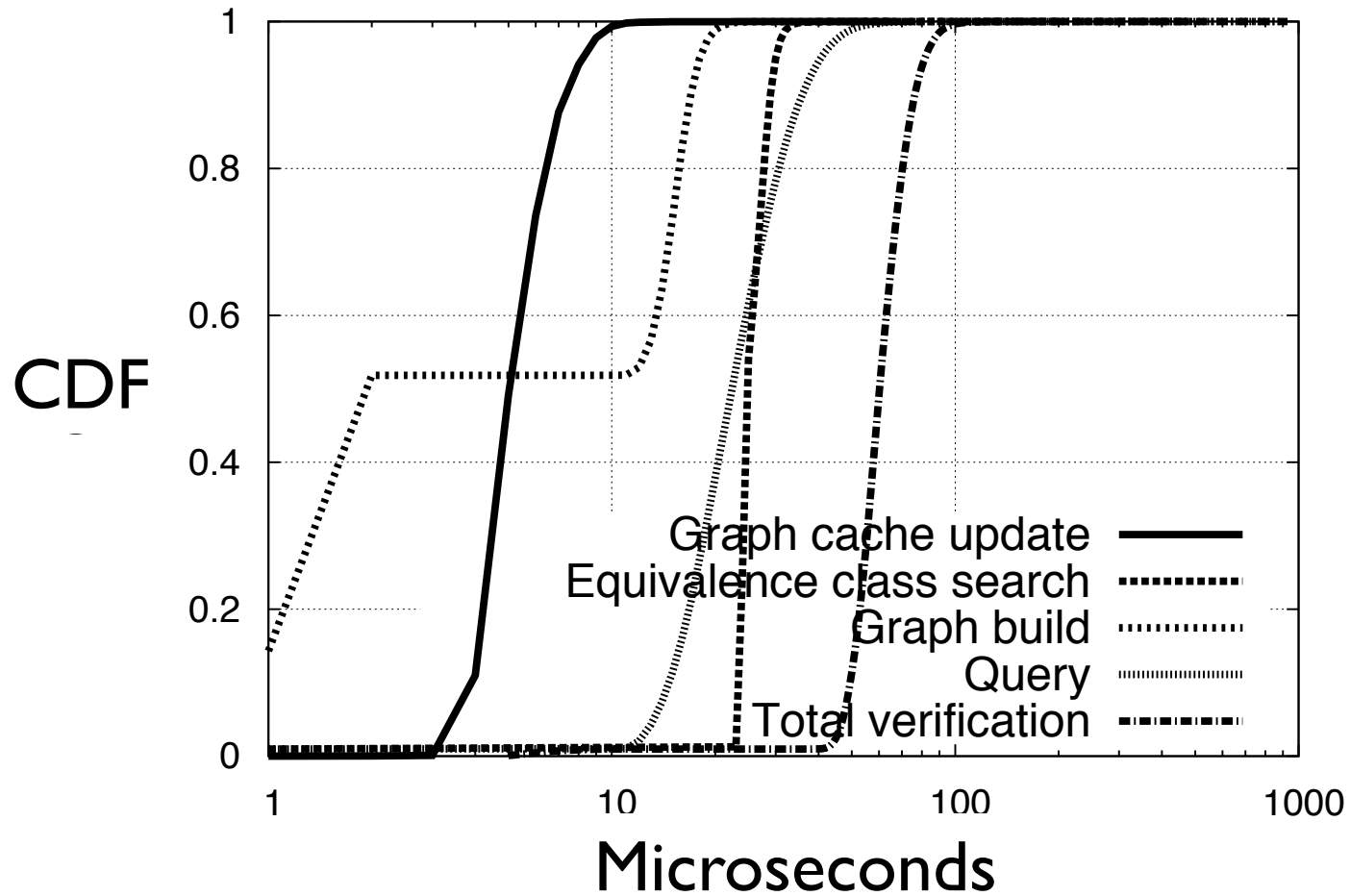


A new experiment not in the paper [with Kelvin Zou]

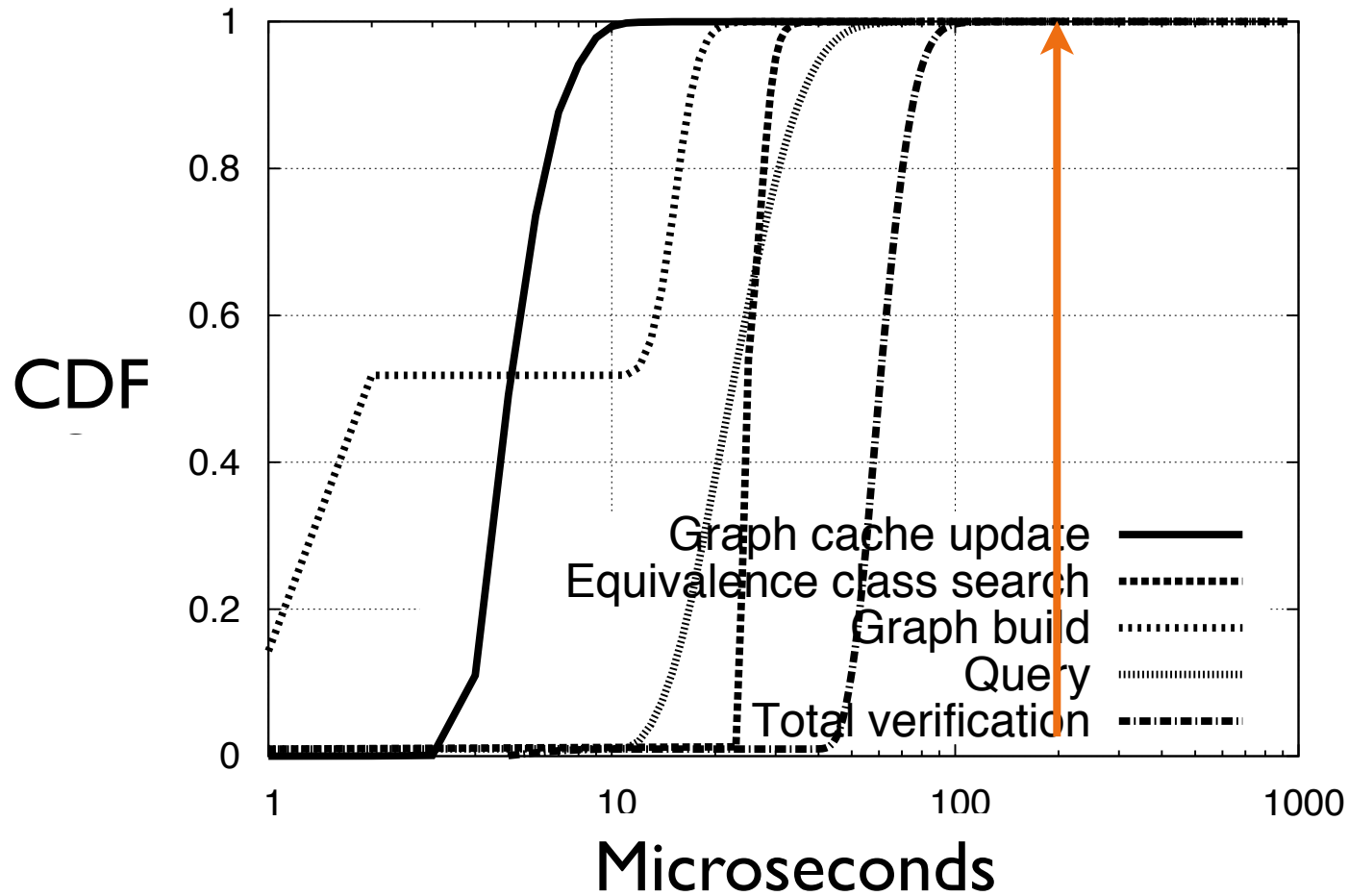
- Mininet OpenFlow network
- 172 switches, 172 hosts
- NOX controller, learning switch app
- TCP connections between random pairs of hosts



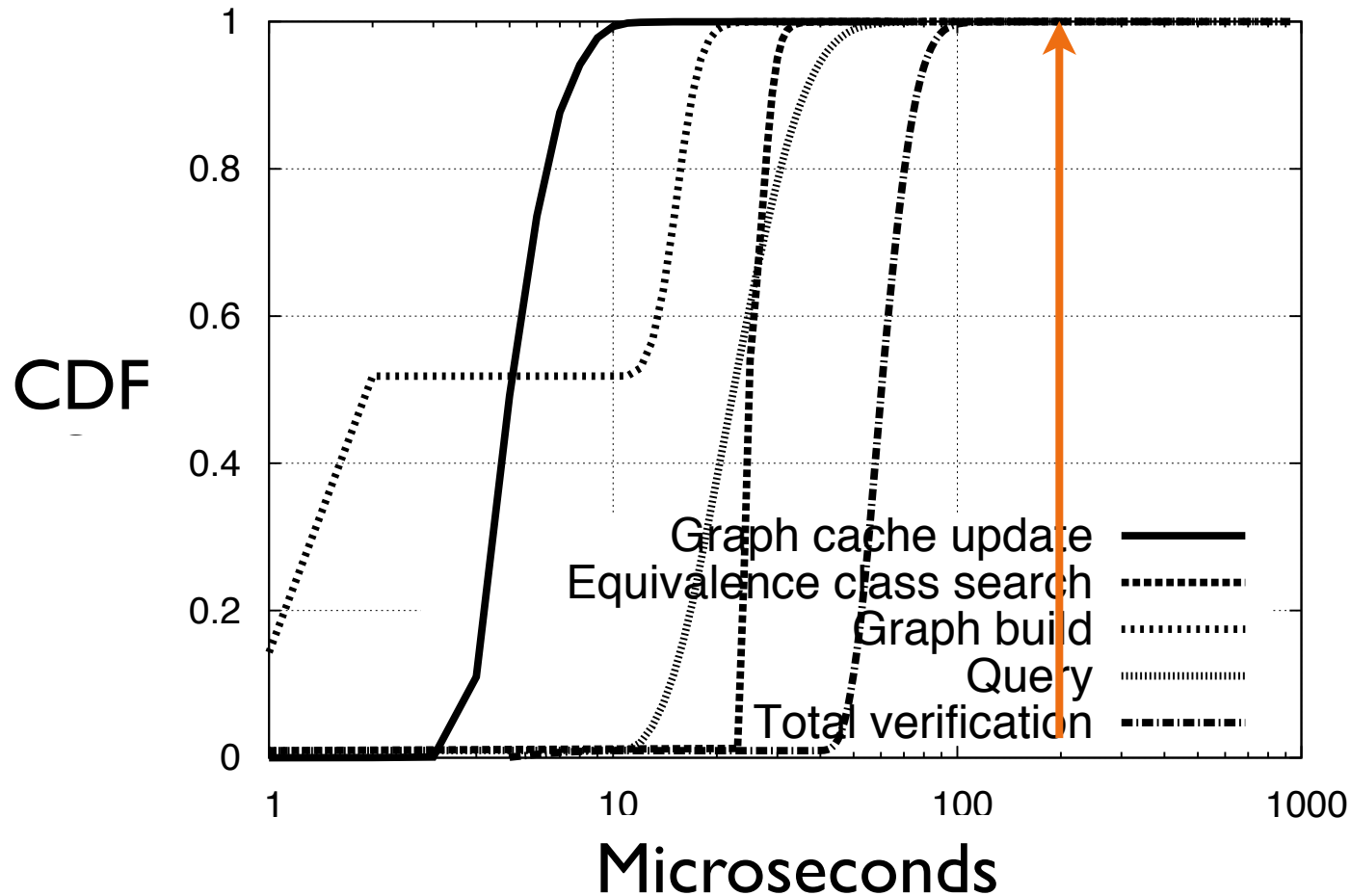
Microbenchmark Runtime



Microbenchmark Runtime



Microbenchmark Runtime



99% of updates verified within 200 μ s



VeriFlow achieves real-time verification

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