A Hypothesis Testing Framework for Network Security

P. Brighten Godfrey
University of Illinois at Urbana-Champaign

TSS Seminar, September 15, 2015

Part of the SoS Lablet with

David Nicol

Kevin Jin

Matthew Caesar

Bill Sanders

Work with...

Anduo Wang

Wenxuan Zhou

Dong Jin

Jason Croft

Matthew Caesar

with

Ahmed Khurshid

Haohui Mai

Xuan Zhou

Rachit Agarwal

Sam King

References to papers in this talk

Haohui Mai, Ahmed Khurshid, Rachit Agarwal, Matthew Caesar, P. Brighten Godfrey, and Samuel T. King. **Debugging the Data Plane with Anteater.** ACM SIGCOMM, August 2011.

Ahmed Khurshid, Xuan Zou, Wenxuan Zhou, Matthew Caesar, and P. Brighten Godfrey. **VeriFlow: Verifying Network-Wide Invariants in Real Time.** 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2013.

Wenxuan Zhou, Dong Jin, Jason Croft, Matthew Caesar, and P. Brighten Godfrey. **Enforcing Customizable Consistency Properties in Software-Defined Networks.** 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2015.

Anduo Wang, Brighten Godfrey, and Matthew Caesar. **Ravel: Orchestrating Software-Defined Networks.** Demo in SOSR'15.

Background:
Network Verification

Networks are complex

89%

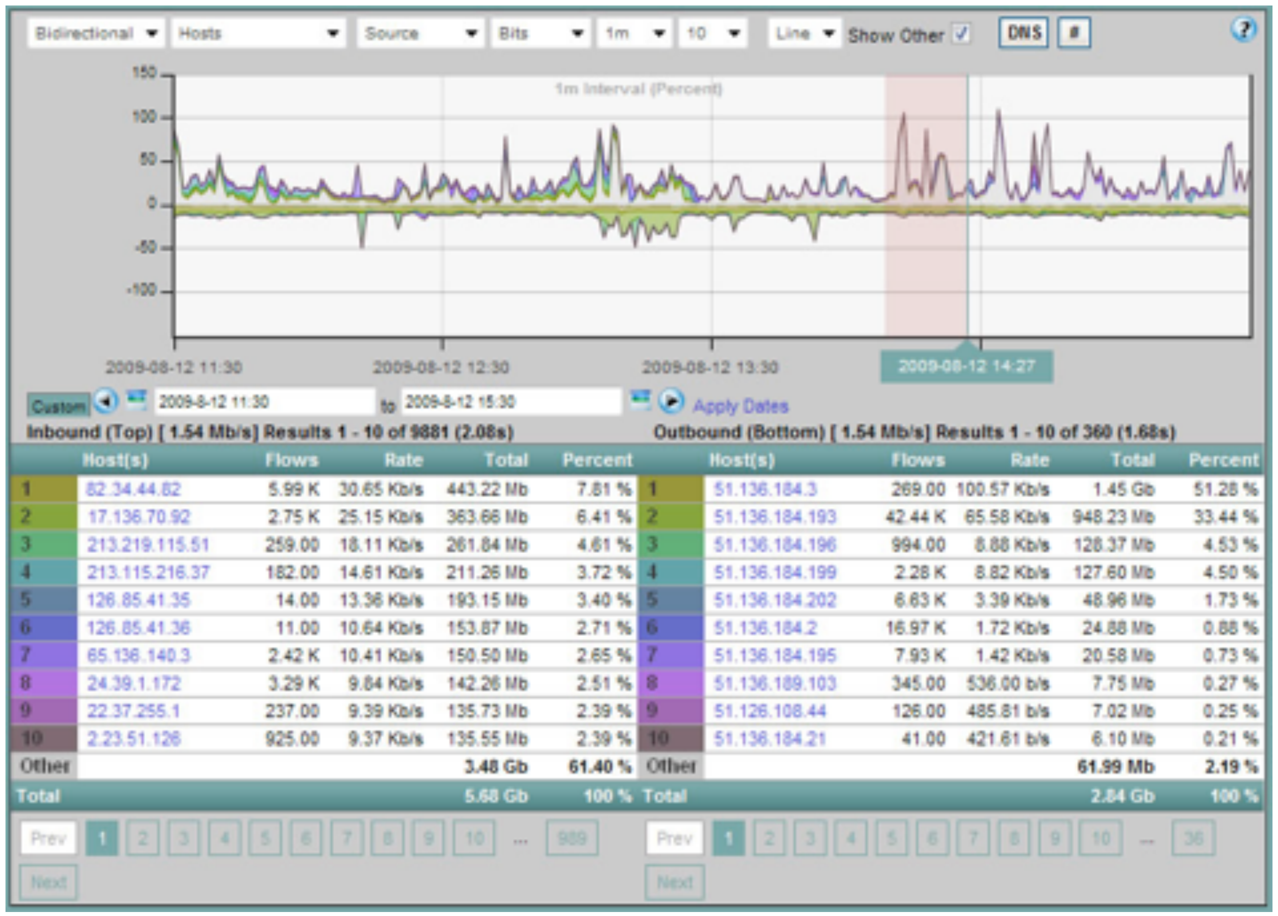
of operators never sure that config changes are bug-free

82%

concerned that changes would cause problems with existing functionality

– Survey of network operators: [Kim, Reich, Gupta, Shahbaz, Feamster, Clark, USENIX NSDI 2015]

Understanding your network




```
hostname bgpdA
password zebra
!
router bgp 8000
  bgp router-id 10.1.4.2

! for the link between A and B
  neighbor 10.1.2.3 remote-as 8000
  neighbor 10.1.2.3 update-source lo0

  network 10.0.0.0/7

! for the link between A and C
  neighbor 10.1.3.3 remote-as 7000
  neighbor 10.1.3.3 ebgp-multihop
  neighbor 10.1.3.3 next-hop-self
  neighbor 10.1.3.3 route-map PP out

! for link between A and D
  neighbor 10.1.4.3 remote-as 6000
  neighbor 10.1.4.3 ebgp-multihop
  neighbor 10.1.4.3 next-hop-self
  neighbor 10.1.4.3 route-map TagD in

! route update filtering
  ip community-list 1 permit 8000:1000
!
```

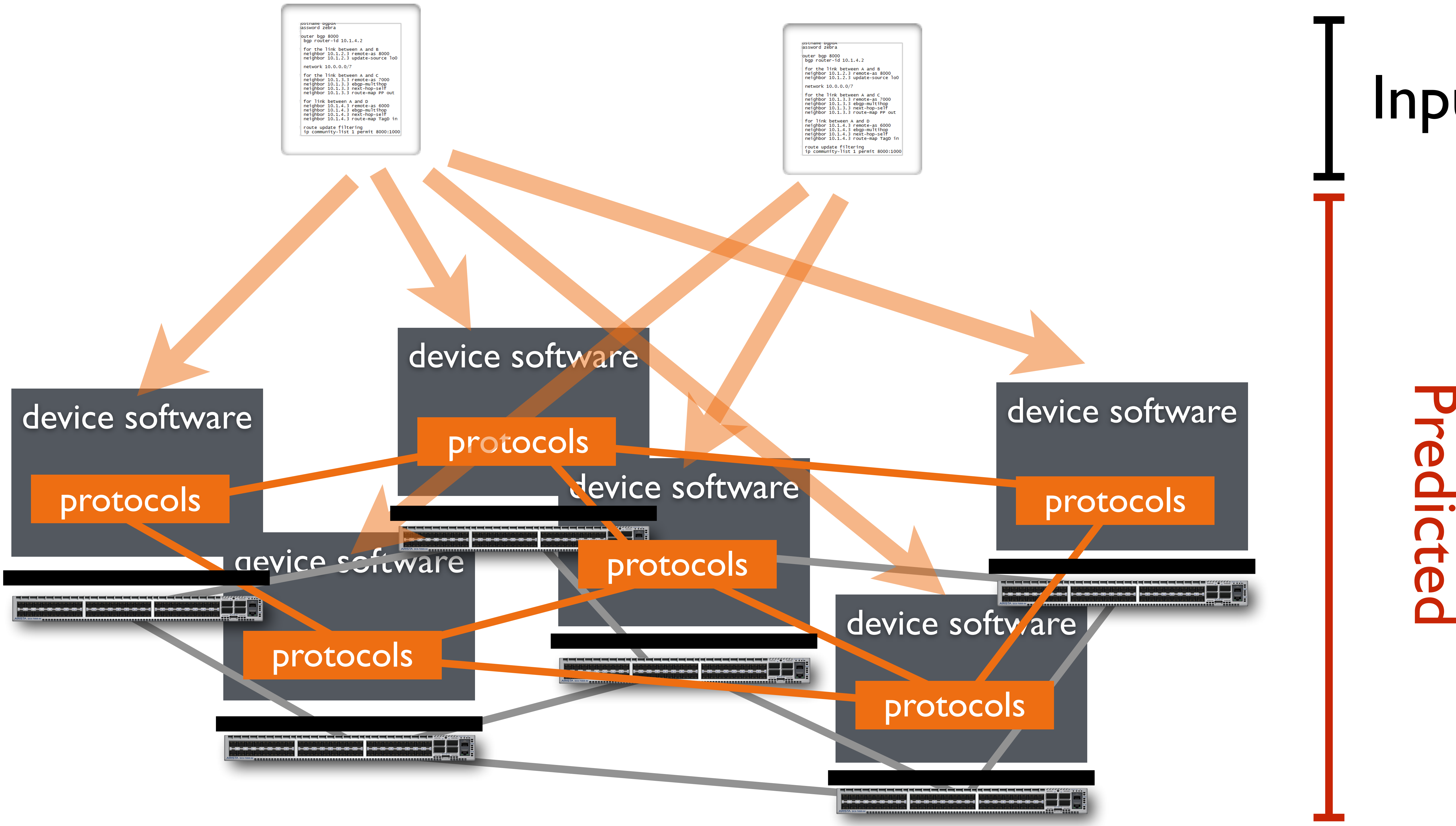
Flow monitoring

Screenshot from Scrutinizer
NetFlow & sFlow analyzer,
snmp.co.uk/scrutinizer/

Configuration verification

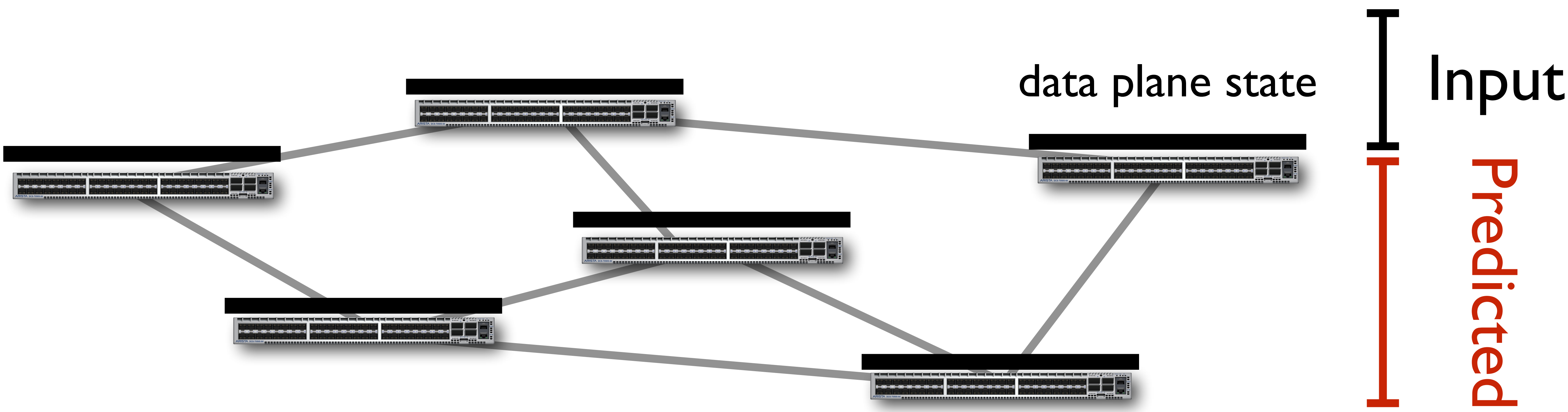
e.g.: RCC for BGP [Feamster & Balakrishnan, NSDI'05]

Configuration verification



Data plane verification

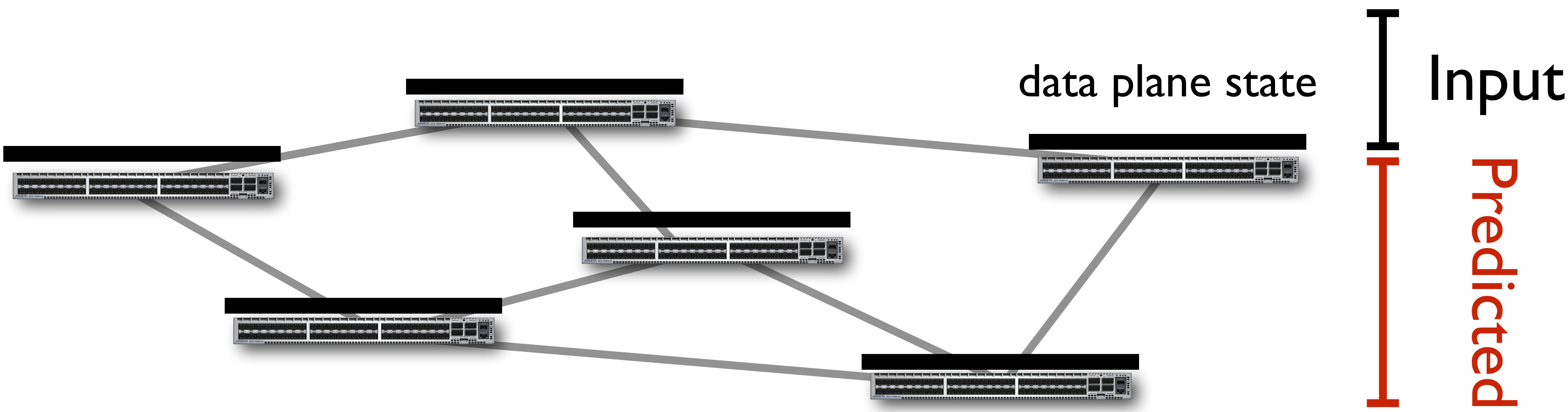
Verify the network
as close as possible to
its actual behavior



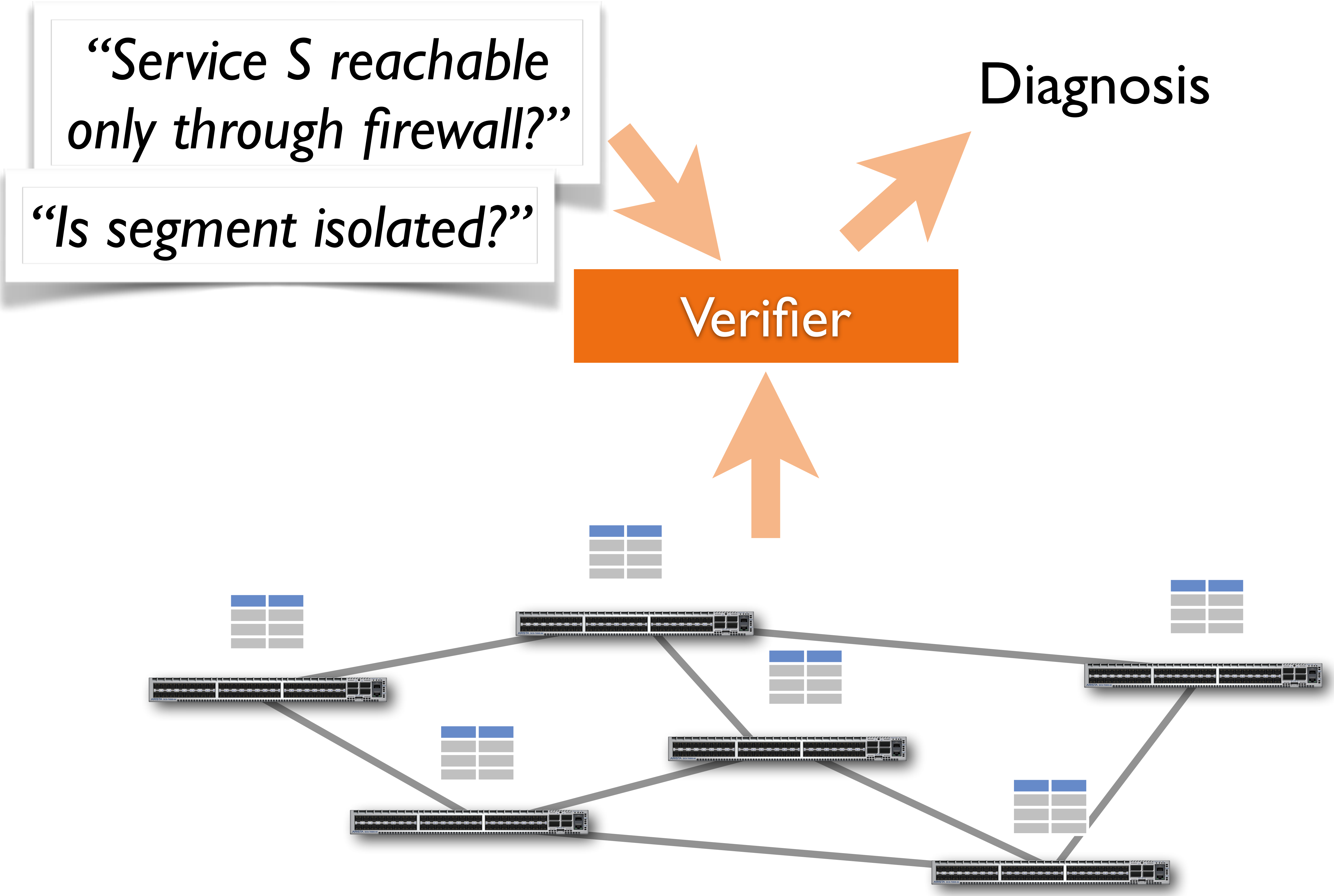
Data plane verification

Verify the network
as close as possible to
its actual behavior

- (Checks current snapshot)
- Insensitive to control protocols
- Accurate model



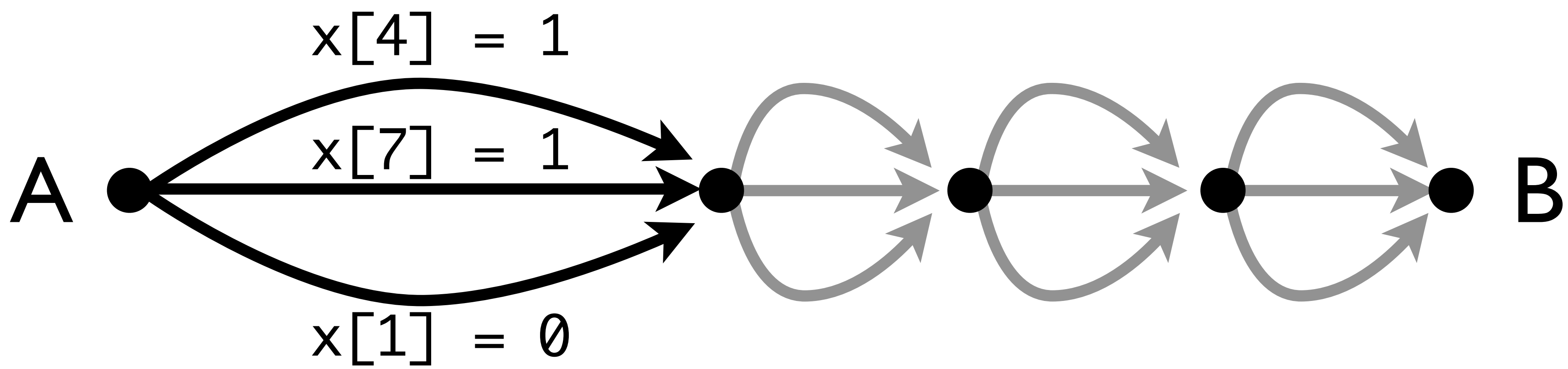
Architecture



Building It

Verification is nontrivial

Packet: `x[0] x[1] x[2] … x[n]`



$$(x_4 \vee x_7 \vee \bar{x_1}) \wedge (\ldots) \wedge (\ldots) \wedge (\ldots)$$

NP-complete!

Anteater's solution

Express data plane and invariants as SAT

- ...up to some max # hops

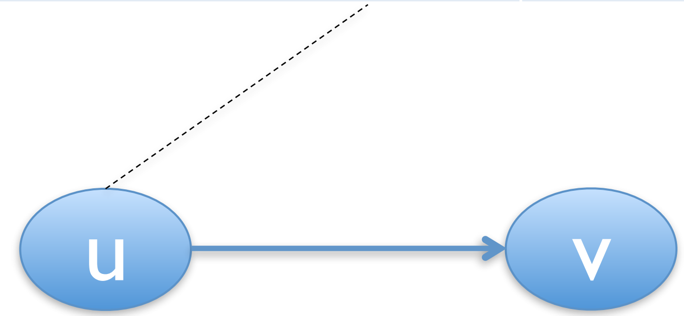
Check with off-the-shelf SAT solver (Boolector)

Data plane as boolean functions

Define P(u, v) as the expression for packets traveling from u to v

- A packet can flow over (u, v) *if and only if* it satisfies P(u, v)

Destination	Action
10.1.1.0/24	Fwd to v



$P(u, v) = dst\_ip \in 10.1.1.0/24$

Reachability as SAT solving

Goal: reachability from u to w

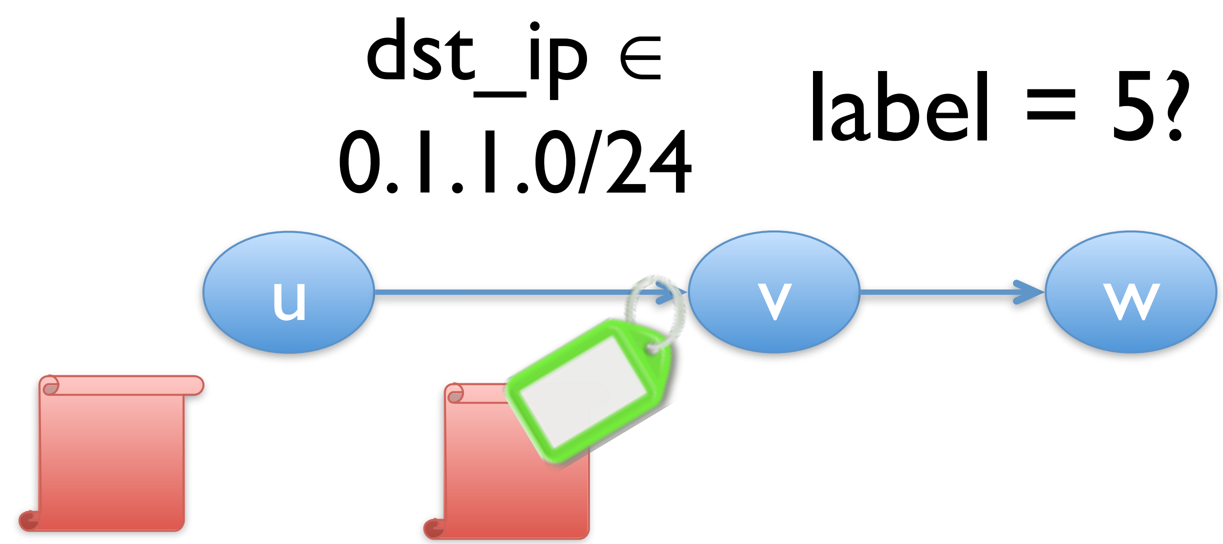
==

$C = (P(u, v) \wedge P(v,w))$ is satisfiable

- SAT solver determines the satisfiability of C
- Problem: exponentially many paths
  - Solution: Dynamic programming (a.k.a. loop unrolling)
  - Intermediate variables: "Can reach *x* in *k* hops?"
  - Similar to [Xie, Zhan, Maltz, Zhang, Greenberg, Hjalmtysson, Rexford, INFOCOM'05]

Packet transformation

Essential to model MPLS, QoS, NAT, etc.



- Model the history of packets: vector over time
- Packet transformation ⇒ boolean constraints over adjacent packet versions

$$(p_i.dst\_ip \in 0.1.1.0/24) \wedge (p_{i+1}.label = 5)$$

More generally: $p_{i+1} = f(p_i)$

Experience with an operational network

Experiences with real network

Evaluated Anteater with operational network

- ∼178 routers supporting >70,000 machines
- Predominantly OSPF, also uses BGP and static routing
- 1,627 FIB entries per router (mean)
- State collected using operator's SNMP scripts

Revealed 23 bugs with 3 invariants in 2 hours

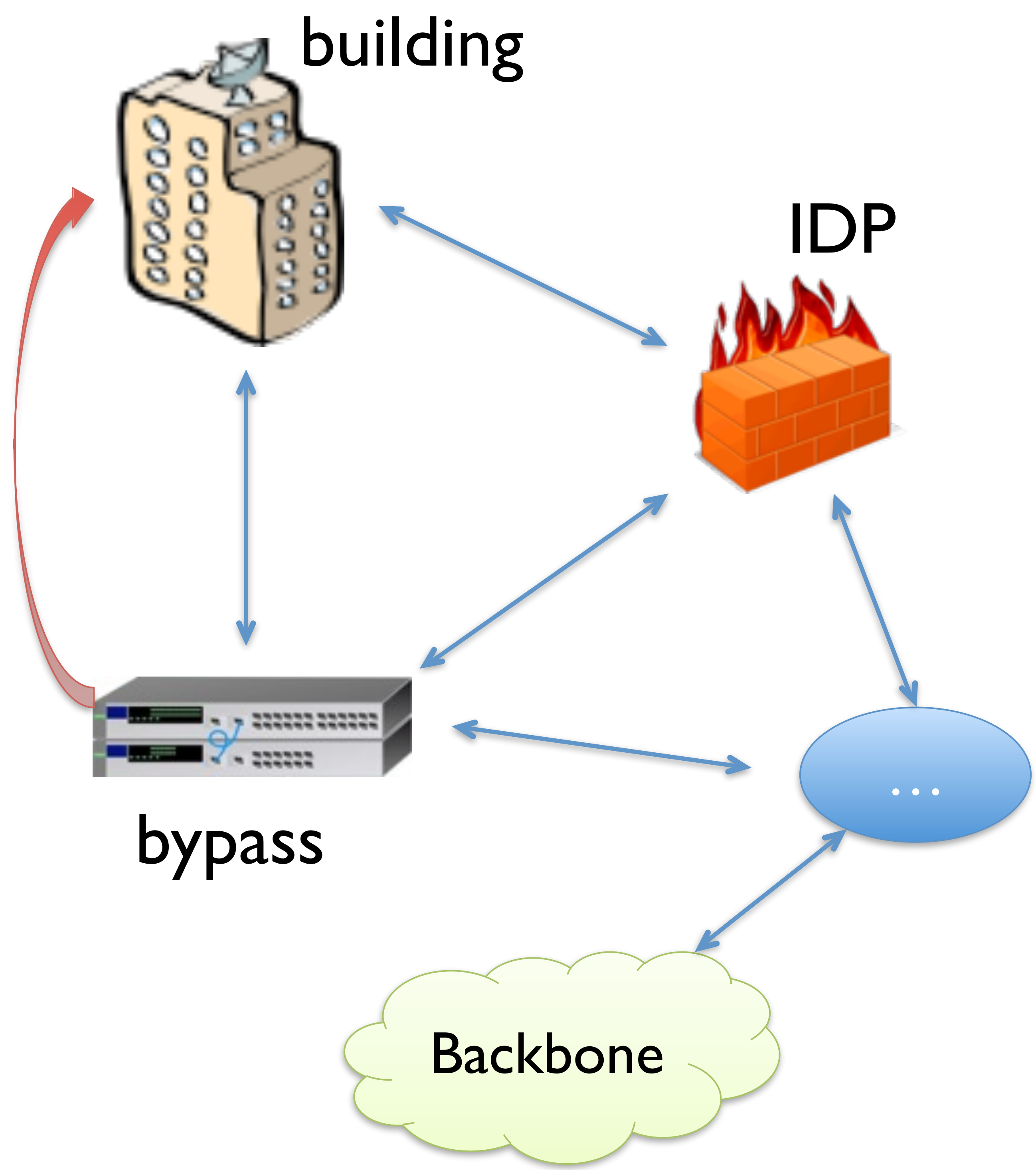
	Loop	Packet loss	Consistency
Being fixed	9	0	0
Stale config.	0	13	1
Total alerts	9	17	2

Forwarding loops

IDP was overloaded, operator introduced bypass

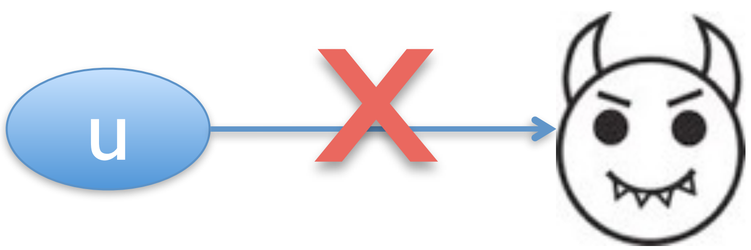
Bypass routed campus traffic to IDP through static routes

Introduced 9 loops



Bugs found by other invariants

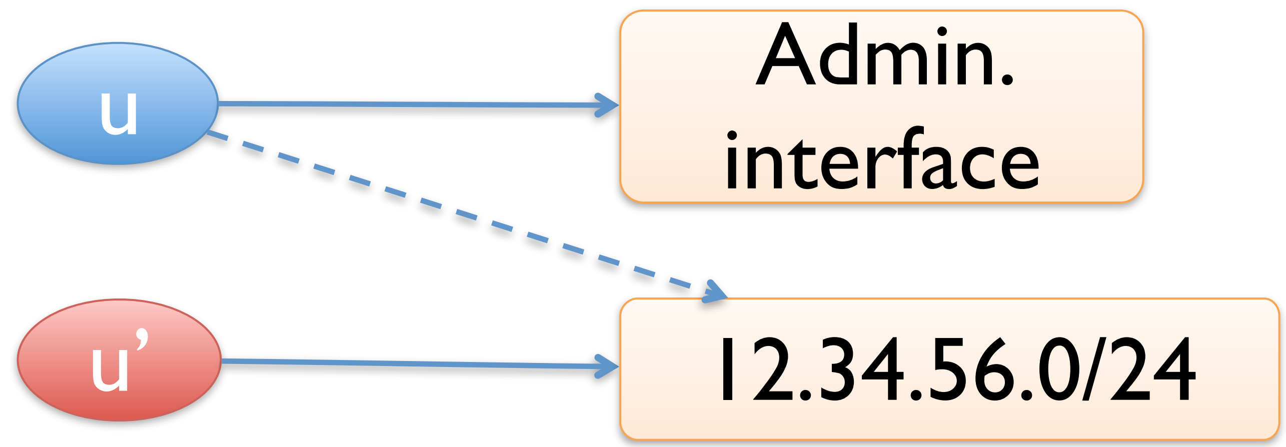
Packet loss



- Blocking compromised machines at IP level
- Stale configuration

  From Sep, 2008

Consistency



- One router exposed web admin interface in FIB
- Different policy on private IP address range

Can we verify networks
in real time?

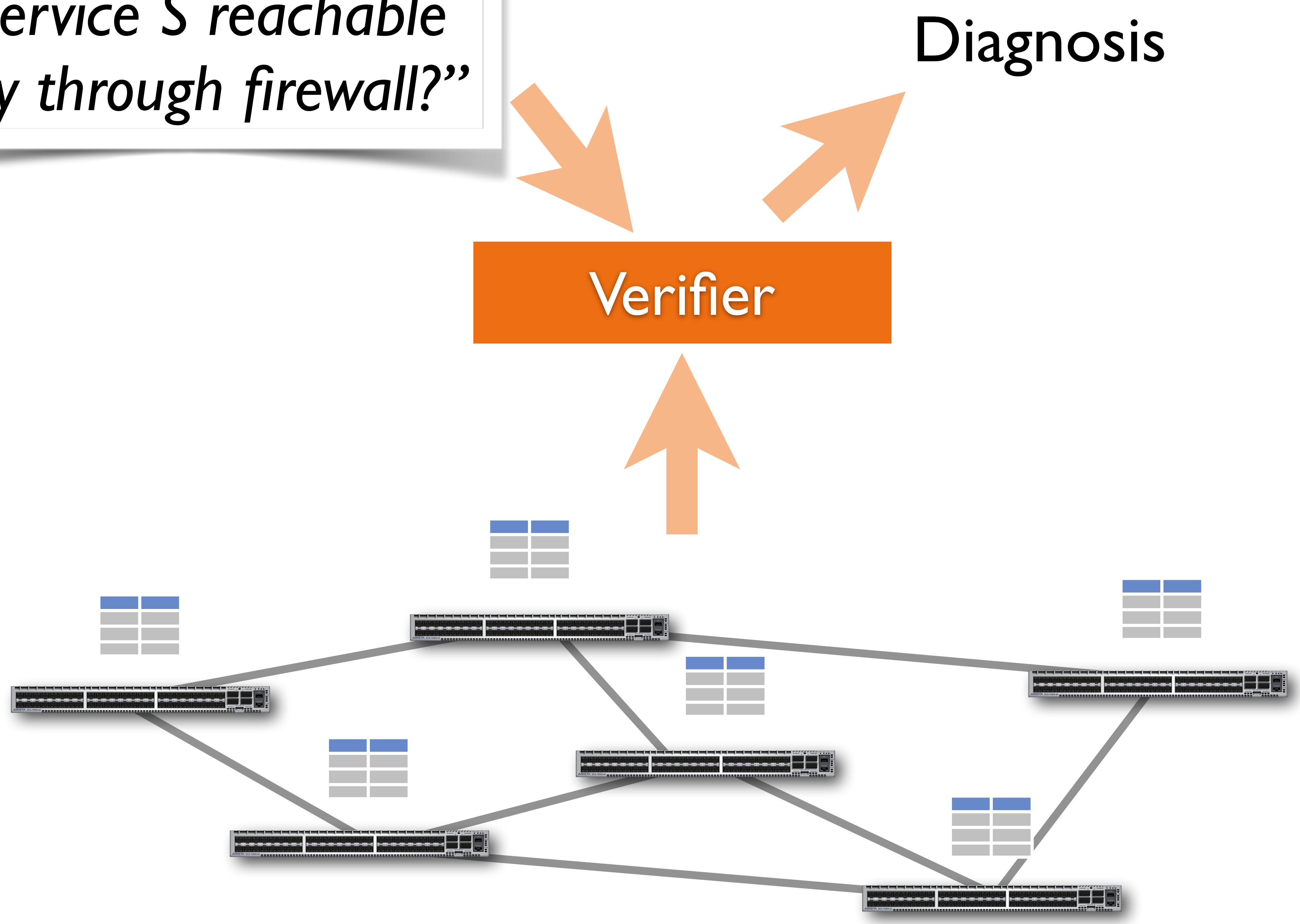
Not so simple

Challenge #1: Obtaining real time view of network

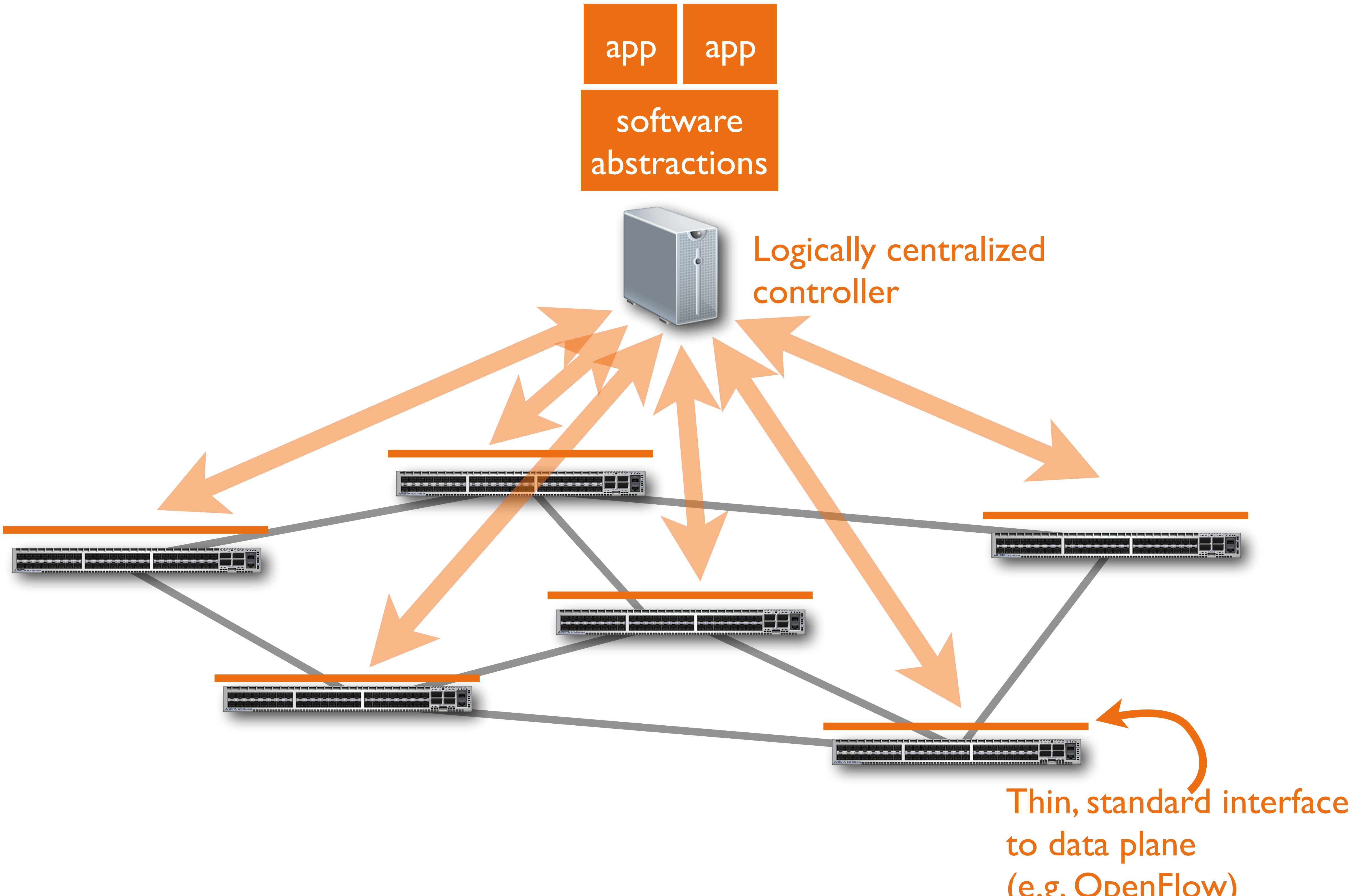
Challenge #2: Verification speed

Architecture

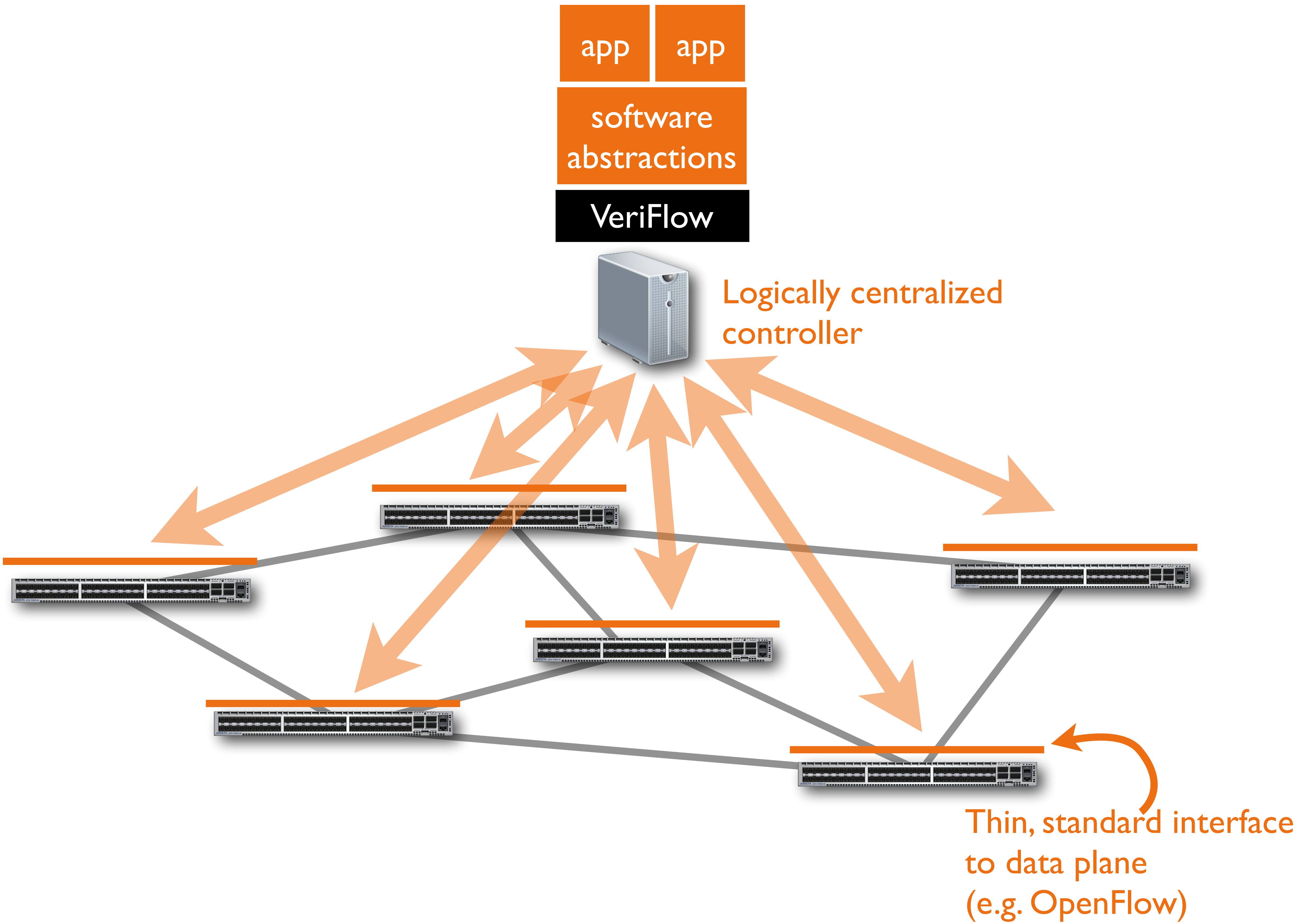
"Service S reachable only through firewall?"



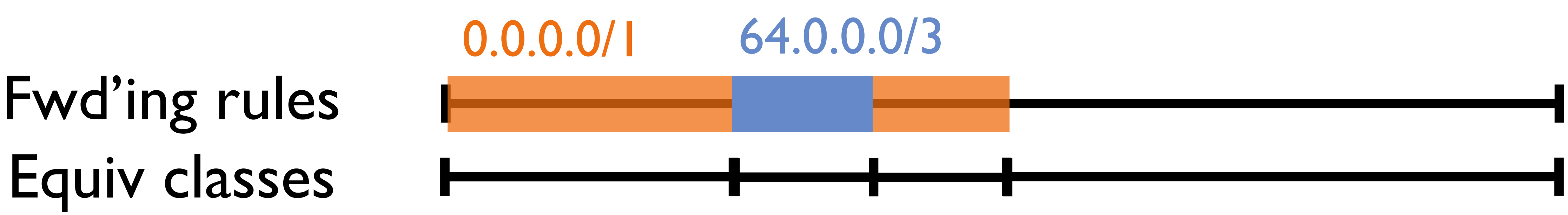
VeriFlow architecture



VeriFlow architecture



Verifying invariants quickly



Find only equivalence classes affected by the update via a multidimensional trie data structure

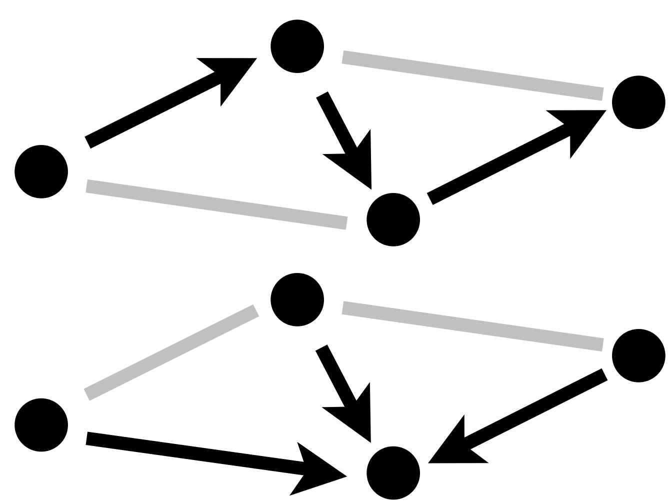
Verifying invariants quickly

Veriflow

Updates

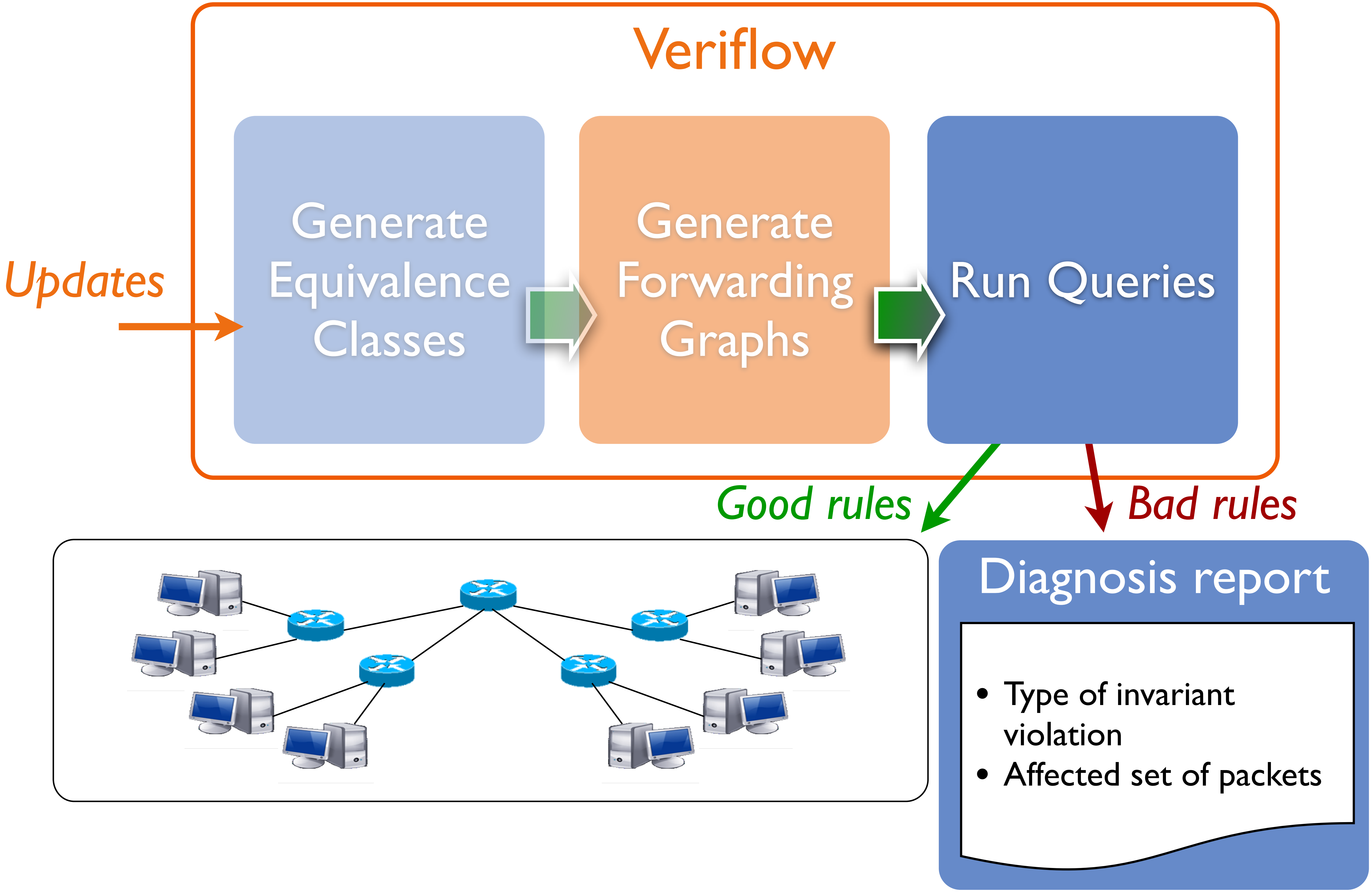
Generate Equivalence Classes

Generate Forwarding Graphs



All the info to answer queries!

Verifying invariants quickly



Evaluation

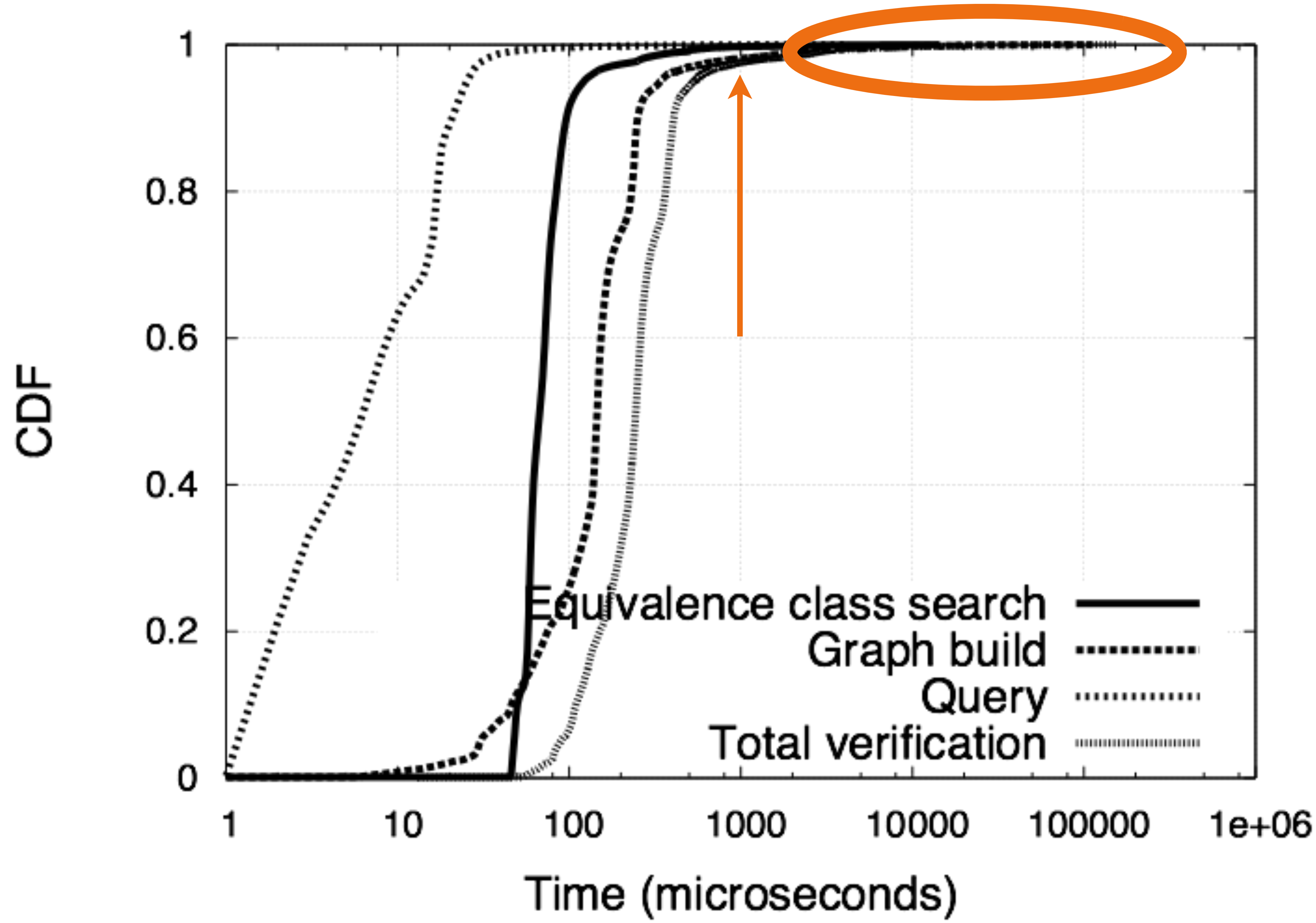
Simulated network

- Real-world BGP routing tables (RIBs) from RouteViews totaling 5 million RIB entries
- Injected into 172-router network (AS 1755 topology)

Measure time to process each forwarding change

- 90,000 updates from Route Views
- Check for loops and black holes

Microbenchmark latency



97.8% of updates verified within 1 ms

Towards a Science of Security:

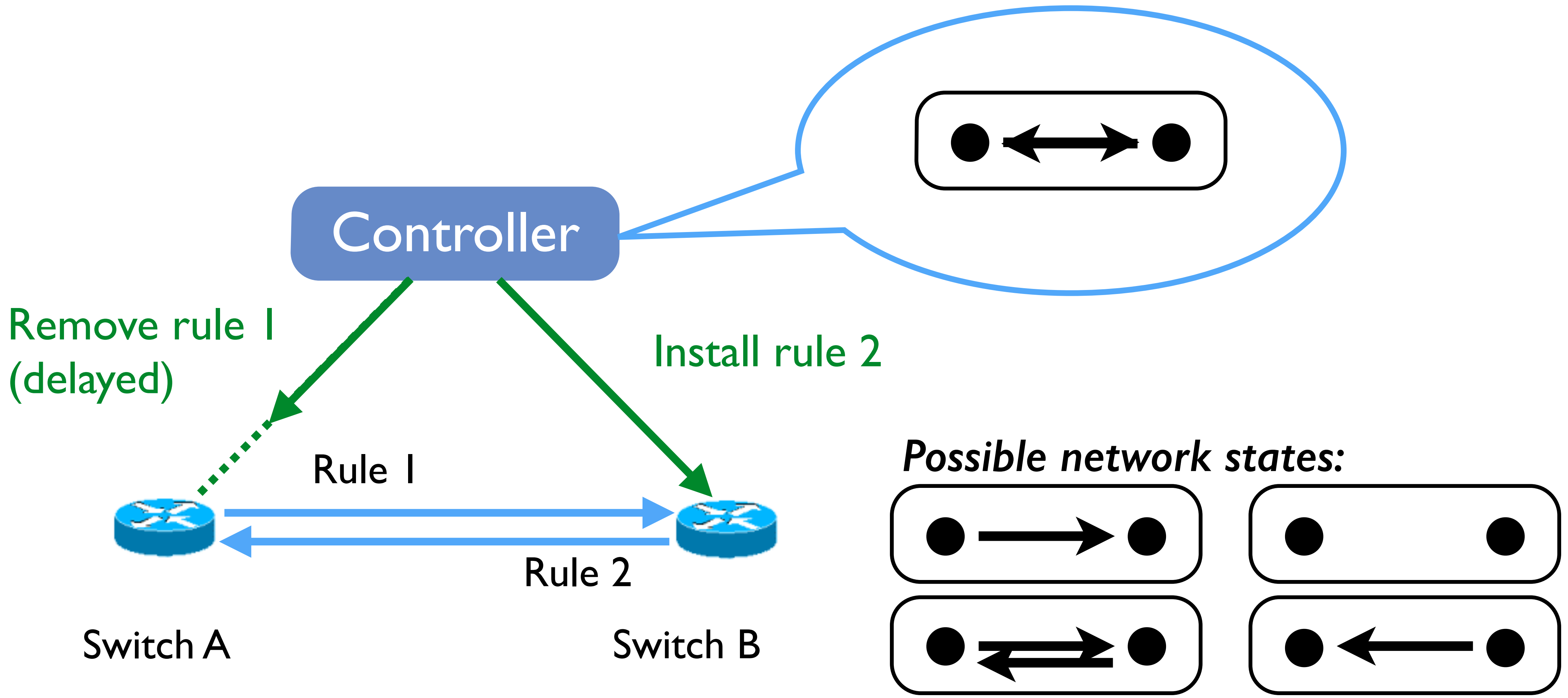
Network Hypothesis Testing

SoS: Network Hypothesis Testing

1. Modeling dynamic networks
2. Networks as databases
3. Provably correct virtual networks

Modeling dynamic networks

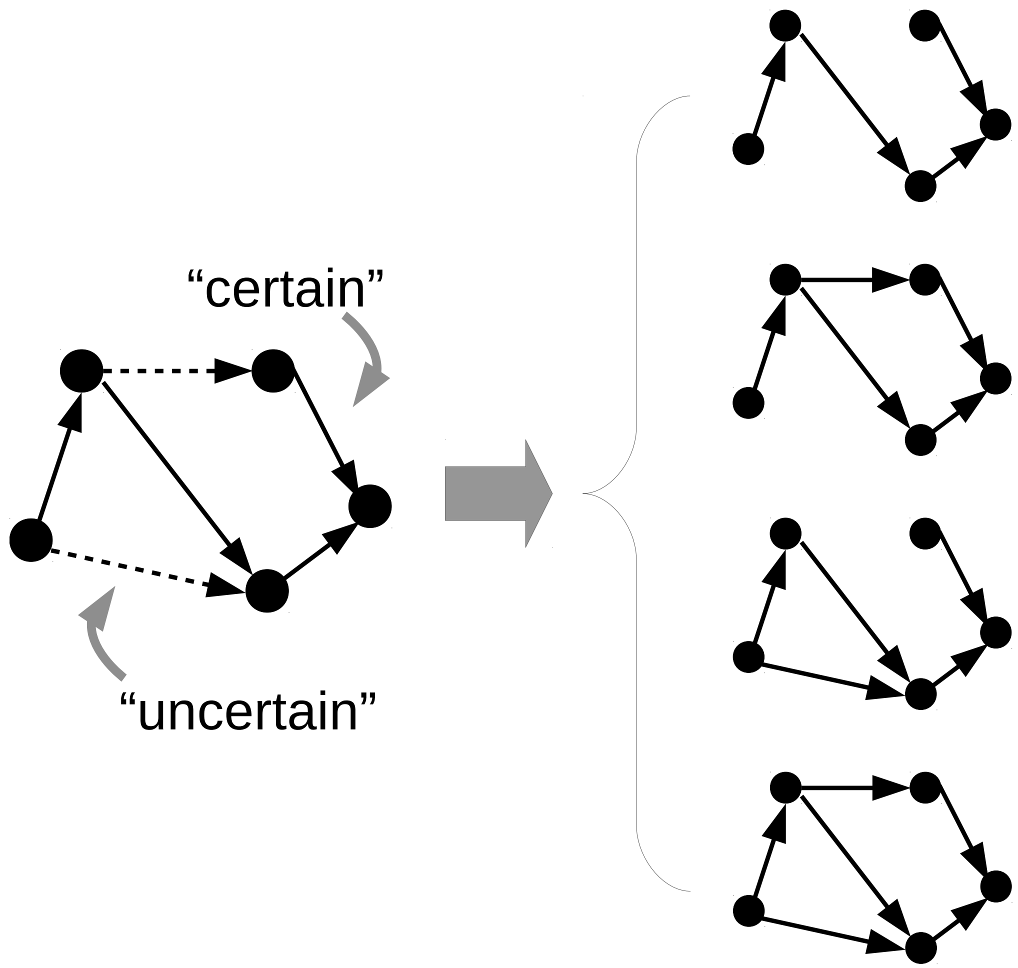
Timing uncertainty



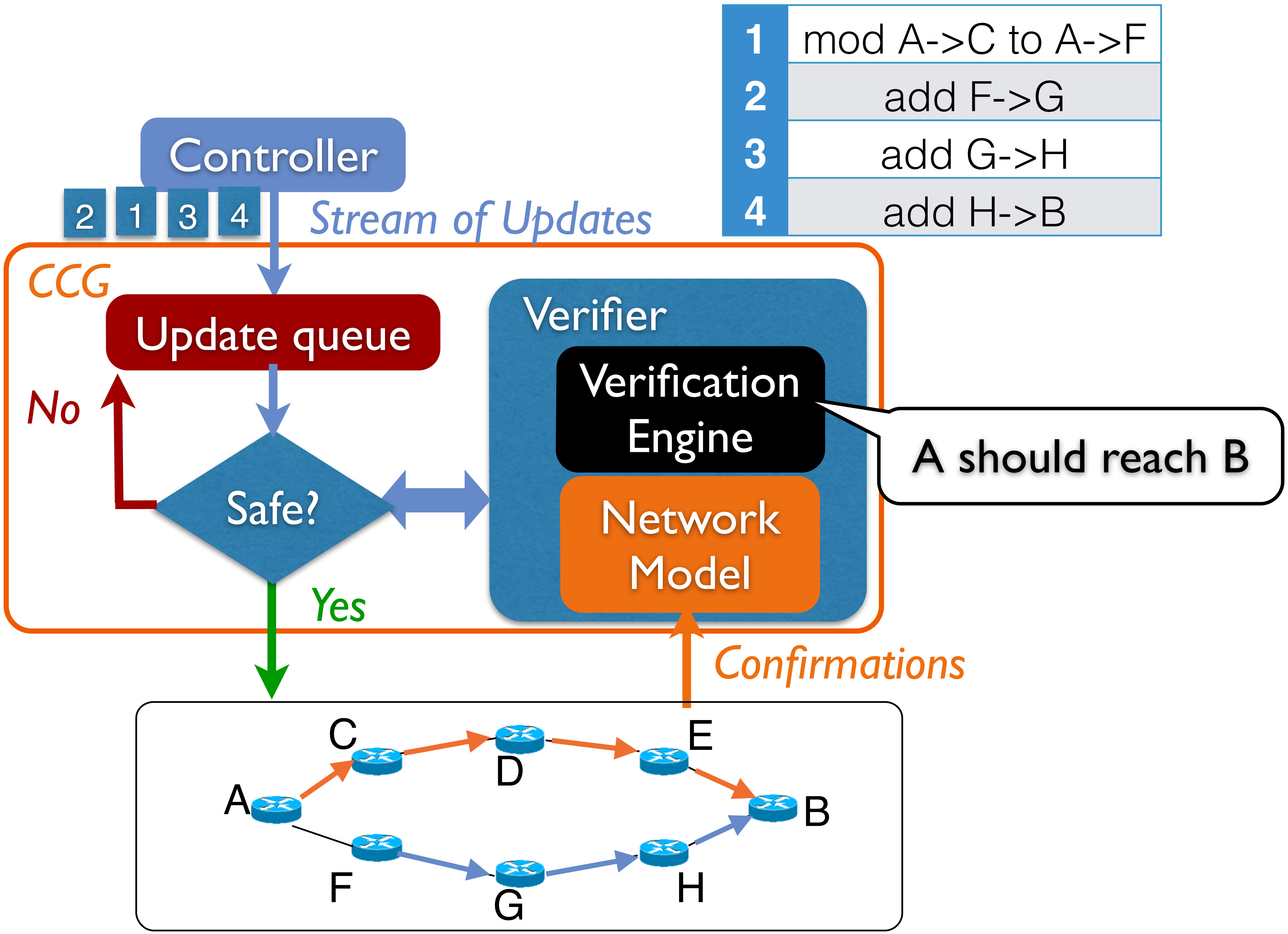
One solution: "consistent updates"

[Reitblatt, Foster, Rexford, Schlesinger, Walker, "Abstractions for Network Update", SIGCOMM 2012]

Uncertainty-aware verification



Update synthesis via verification



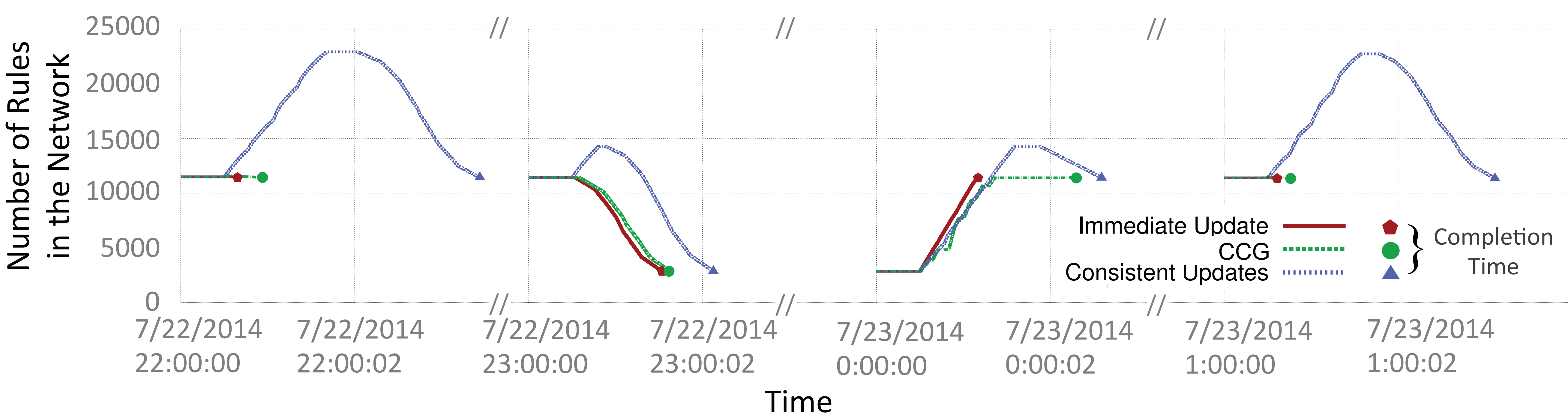
Enforcing dynamic correctness with heuristically maximized parallelism

OK, but…

Can the system "deadlock"?

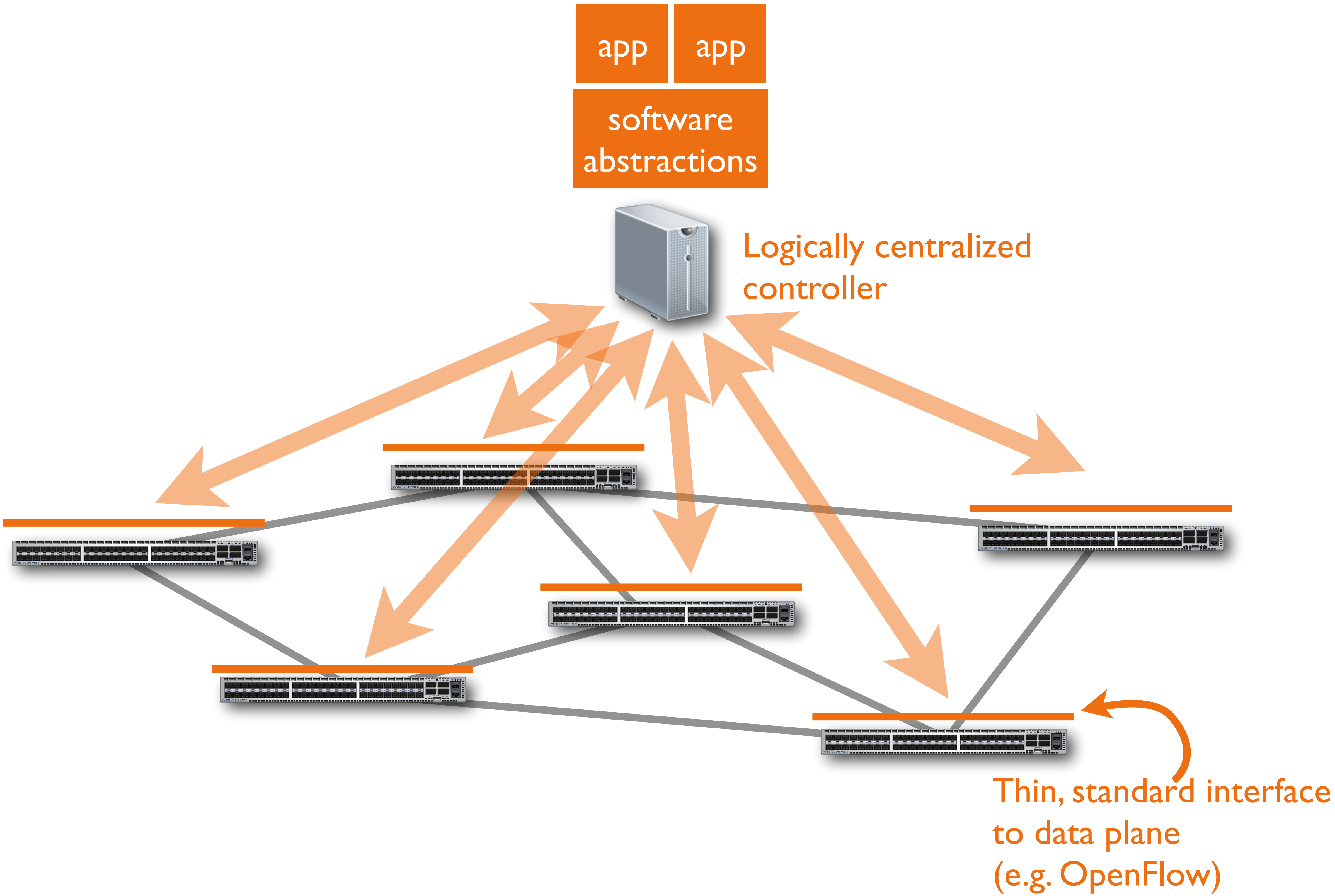
- Proved classes of networks that never deadlock
- Experimentally rare in practice!
- Last resort: heavyweight "fallback" like consistent updates [Reitblatt et al, SIGCOMM 2012]

Is it fast?

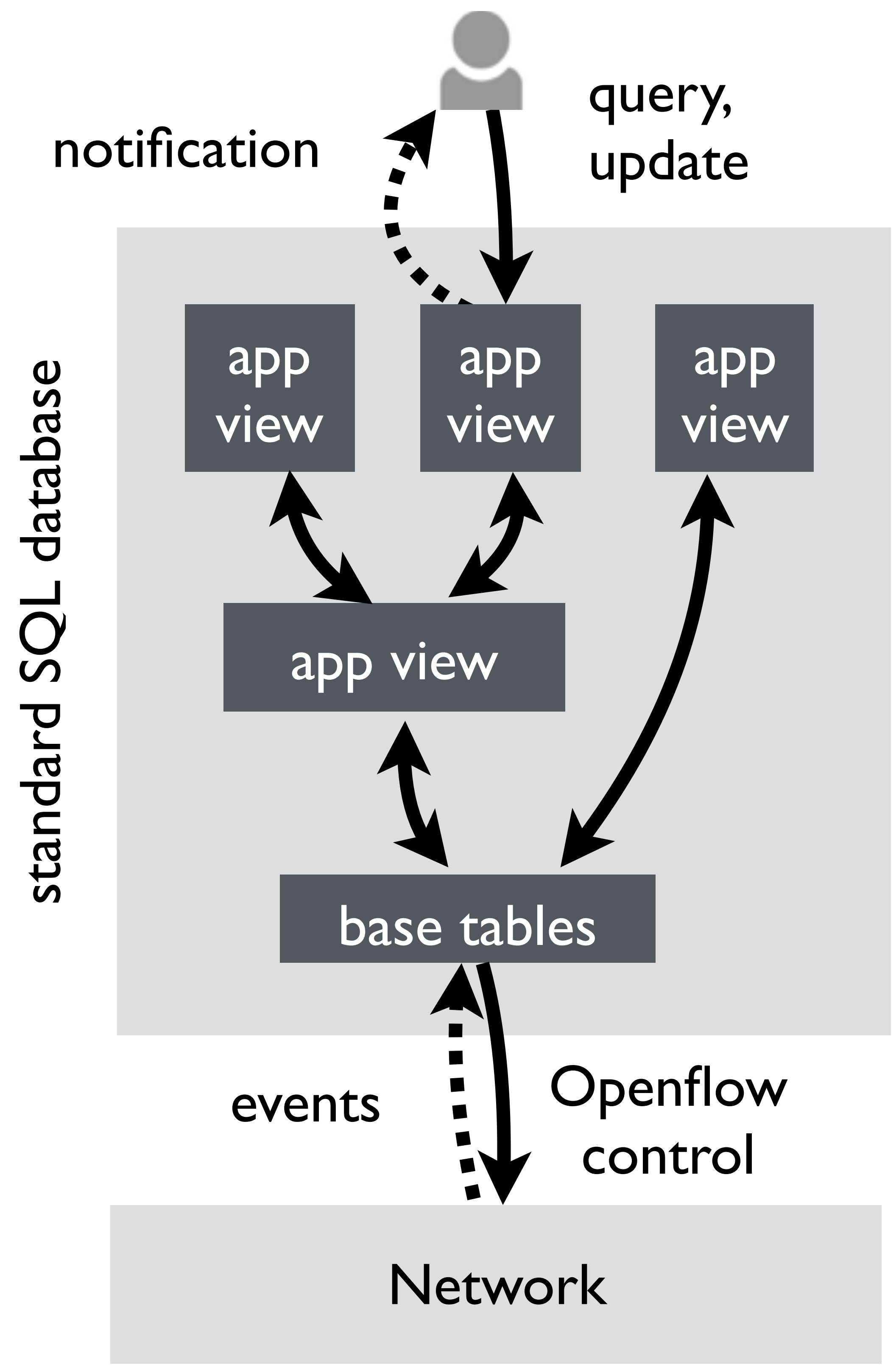


Software-defined Networks as Databases

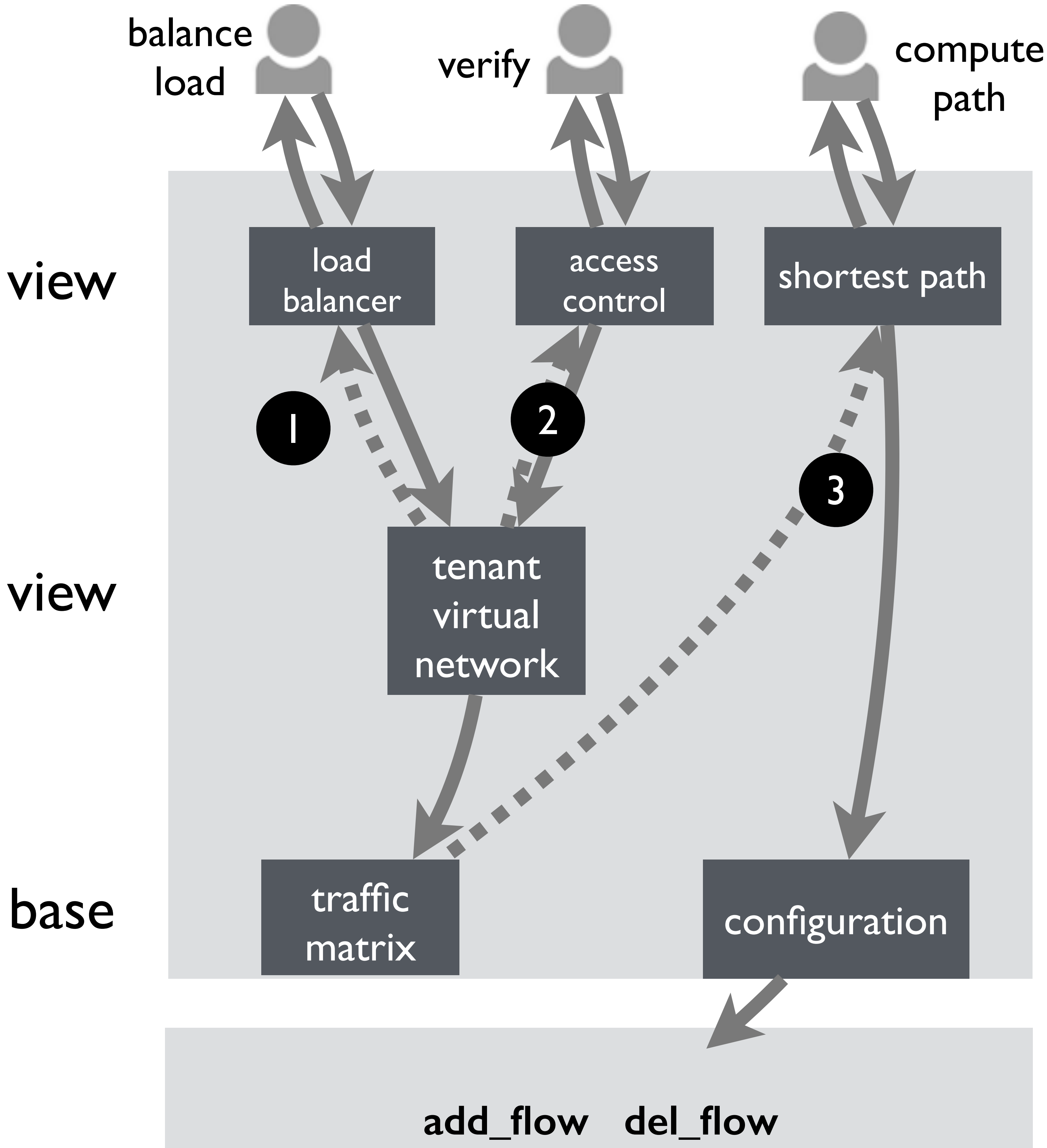
Software-Defined Networks



Ravel: database view of net control



Ravel example



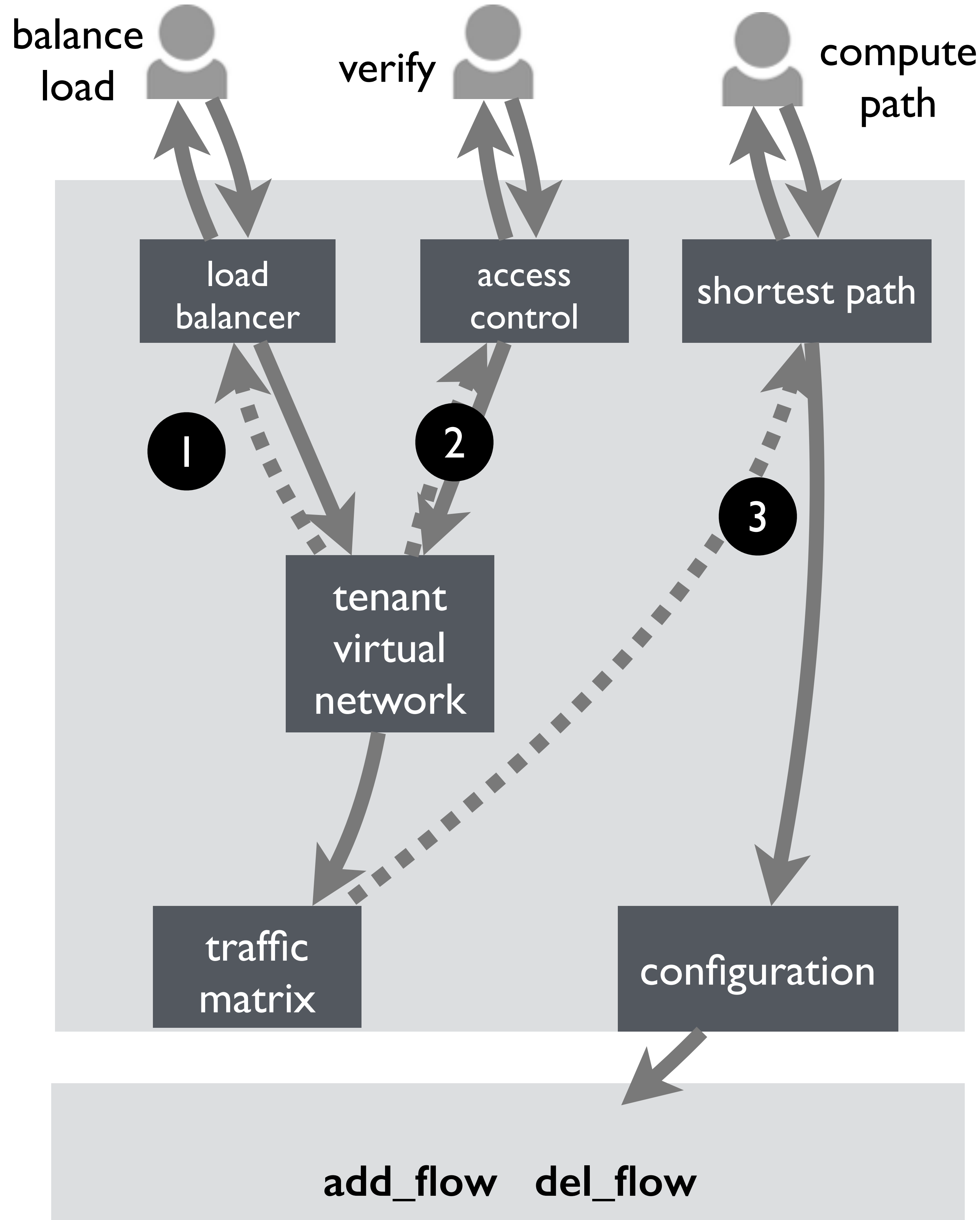
Key benefits

Abstraction via SQL

Orchestration via data-sharing

“Bonus” DB services

- verification, synthesis via view maintenance, update
- transaction processing



Impact of Network Verification

Configuration verification

Configuration verification

- [Al-Shaer2004, Bartal1999, Benson2009, Feamster2005, Yuan2006]

Firewall verification

- Margrave [Nelson, Barratt, Dougherty, Fisler, Krishnamurthi, LISA'10]

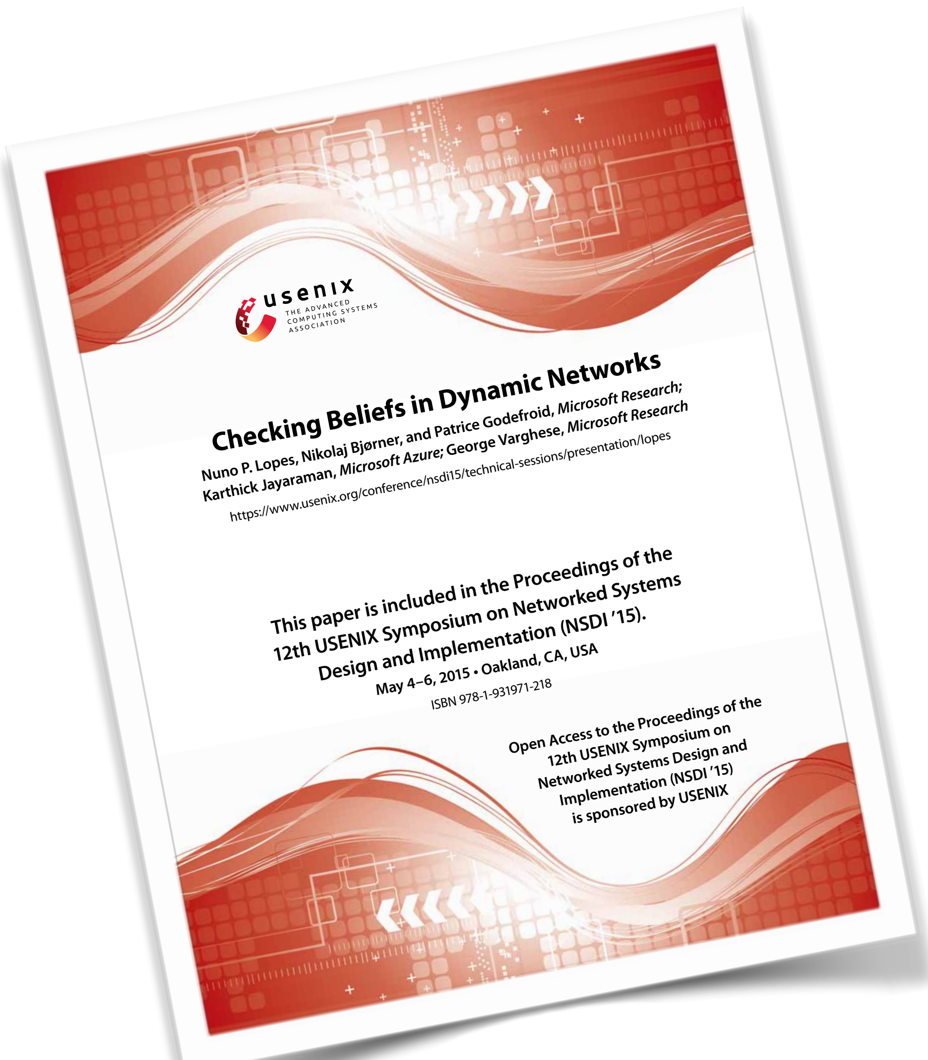
Data plane verification

- Static reachability in IP networks [Xie'05]
- FlowChecker [Al-Shaer, Al-Haj, SafeConfig '10]
- ConfigChecker [Al-Shaer, Al-Saleh, SafeConfig '11]

- Anteater [Mai, Khurshid, Agarwal, Caesar, G., King, SIGCOMM'11]
- VeriFlow [Khurshid, Zou, Zhou, Caesar, G., HotSDN'12, NSDI'13]
- CCG [Zhou, Jin, Croft, Caesar, G., NSDI'15]

- Header Space Analysis [Kazemian, Varghese, and McKeown, NSDI '12]
- NetPlumber [Kazemian, Chang, Zeng, Varghese, McKeown, Whyte, NSDI '13]
- Batfish [Fogel, Fung, Pedrosa, Walraed-Sullivan, Govindan, Mahajan, Millstein, NSDI'15]

DPV in the real world


Checking Beliefs in Dynamic Networks

Nuno P. Lopes, Nikolaj Bjørner, and Patrice Godefroid, *Microsoft Research;*
Karthick Jayaraman, *Microsoft Azure;* George Varghese, *Microsoft Research*

https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/lopes

This paper is included in the Proceedings of the
12th USENIX Symposium on Networked Systems
Design and Implementation (NSDI '15).
May 4–6, 2015 • Oakland, CA, USA

ISBN 978-1-931971-218

Open Access to the Proceedings of the
12th USENIX Symposium on
Networked Systems Design and
Implementation (NSDI '15)
is sponsored by USENIX

Microsoft

veriflow
SYSTEMS

Future research: Richer models

Software pipelines

Stateful Networks

Verifiable SDN Controllers

Higher layer concepts
(roles, people, applications)

Thanks!