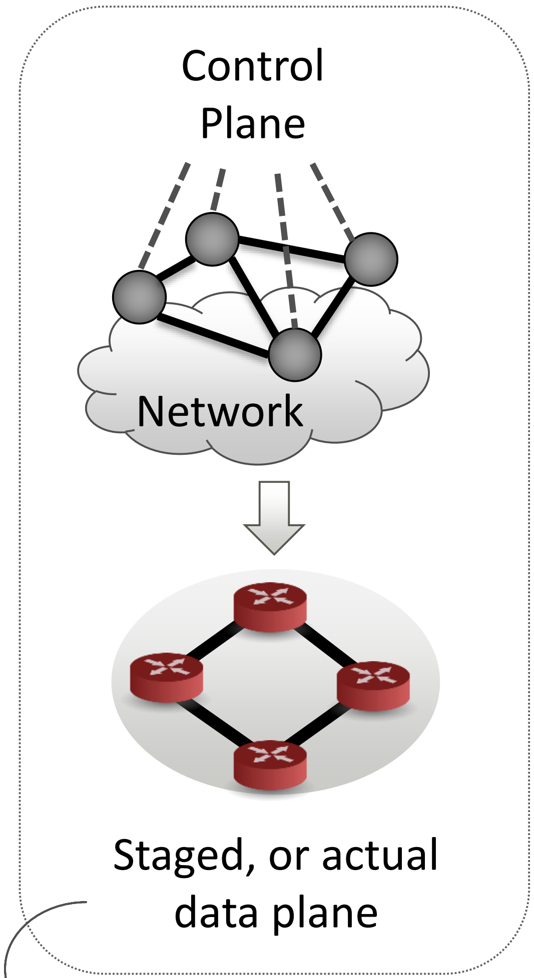
Delta-net: Real-time Network Verification Using Atoms

Alex Horn[1], Ali Kheradmand[2] and Mukul R. Prasad[1]

[1] Fujitsu Labs of America
[2] University of Illinois at Urbana-Champaign (Internship)

NSDI 2017, Boston, March 29

Context: “Network Verification”

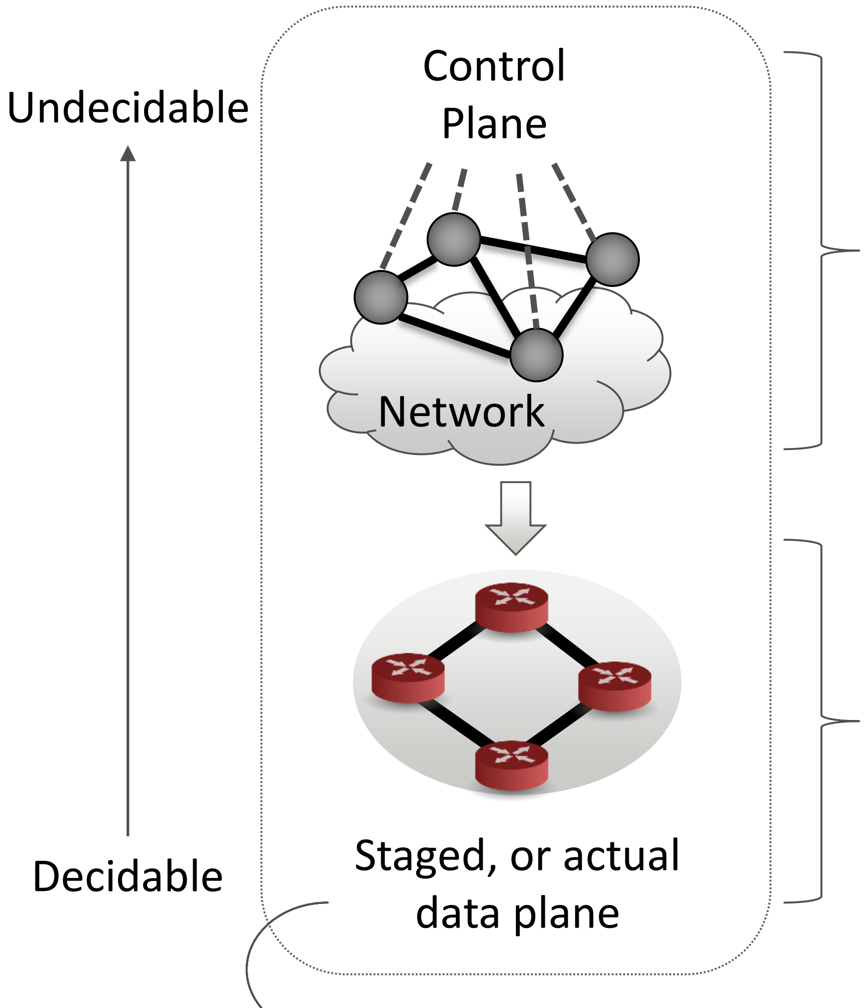


- [Canini et al., NSDI 2012], ***NICE***
- [Ball et al., PLDI 2014], ***VeriCon***
- [Fayaz et., NSDI 2016], ***BUZZ***
- [Stoenescu et al., SIGCOMM 2016], ***SymNet***
- [Fogel et al., NSDI 2015], ***Batfish***
- [Lopes et al., NSDI 2015], ***NoD in Z3***
- …

- [Khurshid et al., NSDI 2013], ***Veriflow***
- [Kazemian et al., NSDI 2013], ***NetPlumber***
- [Yang and Lam, ICNP 2013], ***AP Verifier***
- …

Goal: detect network outages before they occur

Context: “Network Verification”

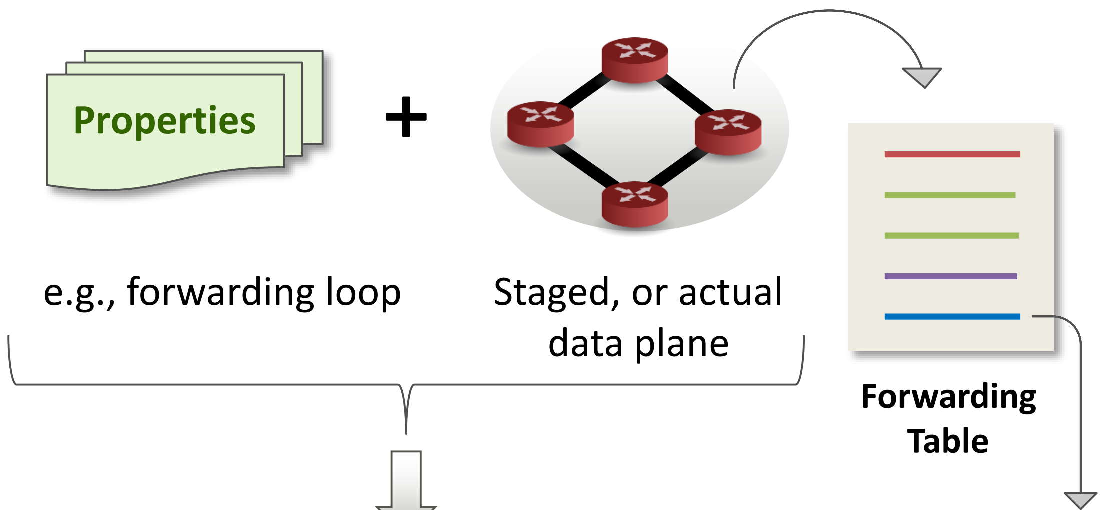


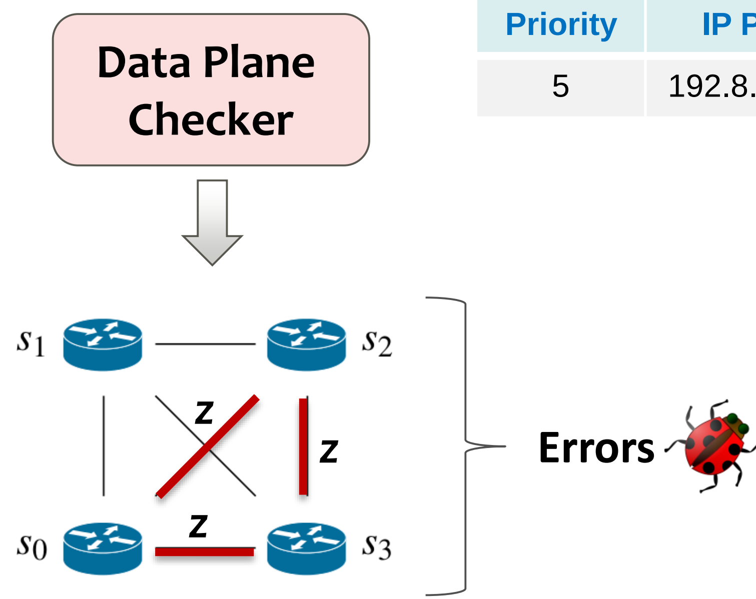
- [Canini et al., NSDI 2012], ***NICE***
- [Ball et al., PLDI 2014], ***VeriCon***
- [Fayaz et., NSDI 2016], ***BUZZ***
- [Stoenescu et al., SIGCOMM 2016], ***SymNet***
- [Fogel et al., NSDI 2015], ***Batfish***
- [Lopes et al., NSDI 2015], ***NoD in Z3***
- …

- [Khurshid et al., NSDI 2013], ***Veriflow***
- [Kazemian et al., NSDI 2013], ***NetPlumber***
- [Yang and Lam, ICNP 2013], ***AP Verifier***
- …

Goal: detect network outages before they occur

Big Picture of Real-time Network Verification

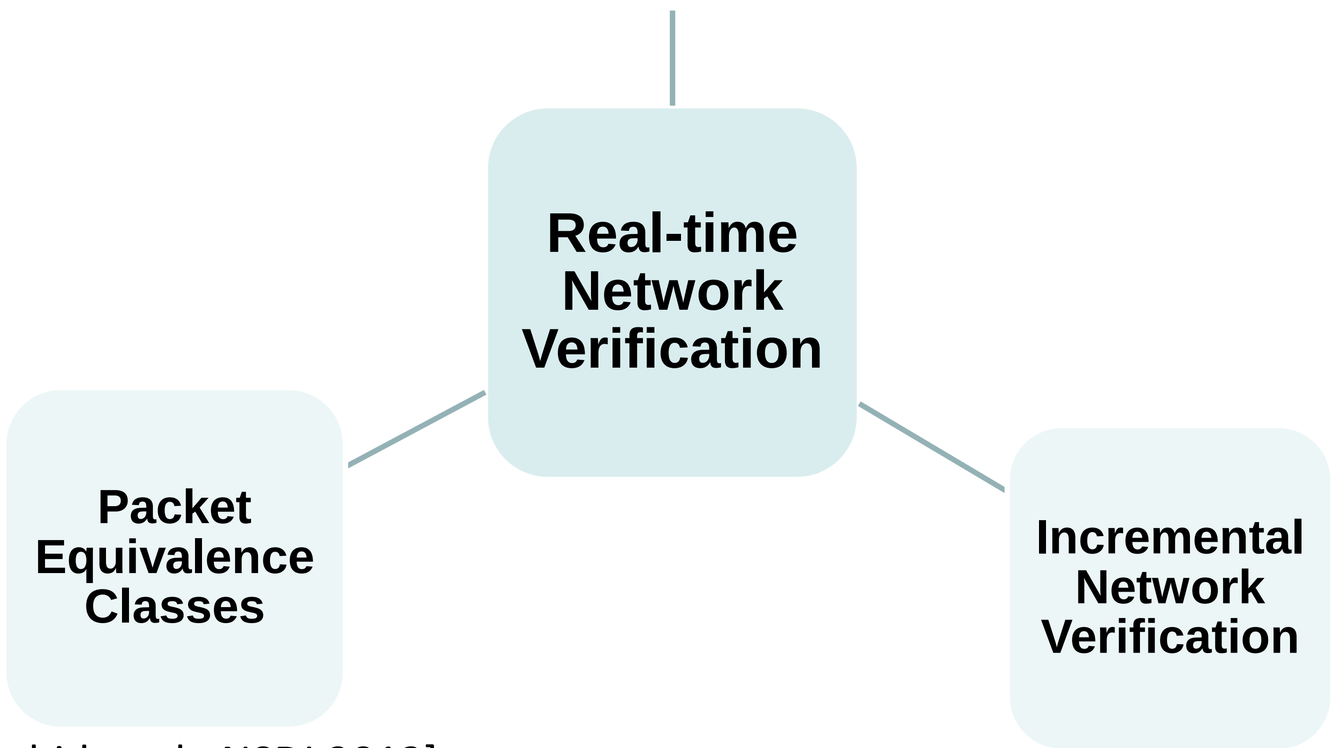




Priority	IP Prefix	Action	Next Hop
5	192.8.0.10/31	Forward	123.9.1.12

Forwarding Rule

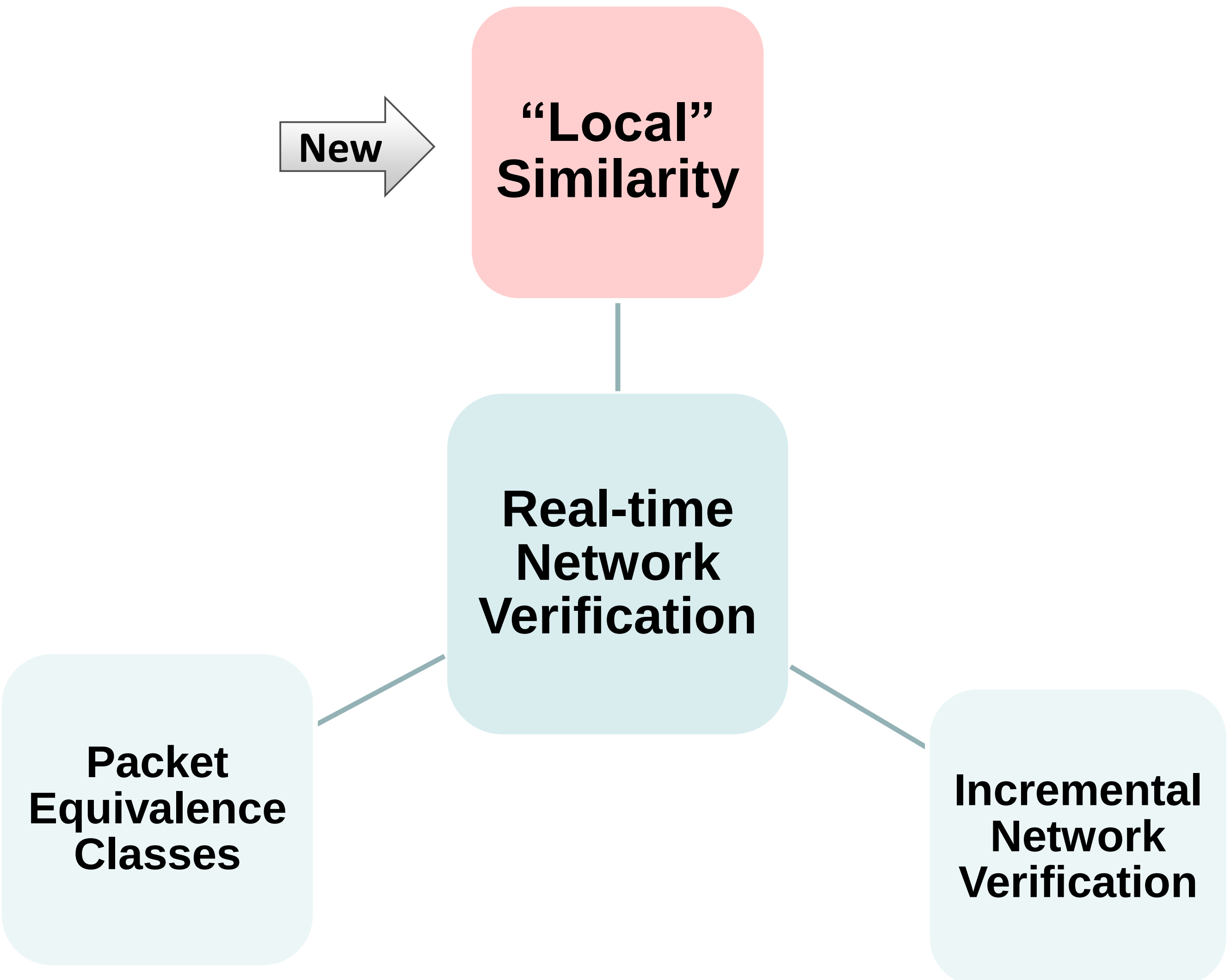
Taxonomy



[Khurshid et al., NSDI 2013]
[Kazemian et al., NSDI 2013]
[Yang and Lam, ICNP 2013]

[Khurshid et al., NSDI 2013]

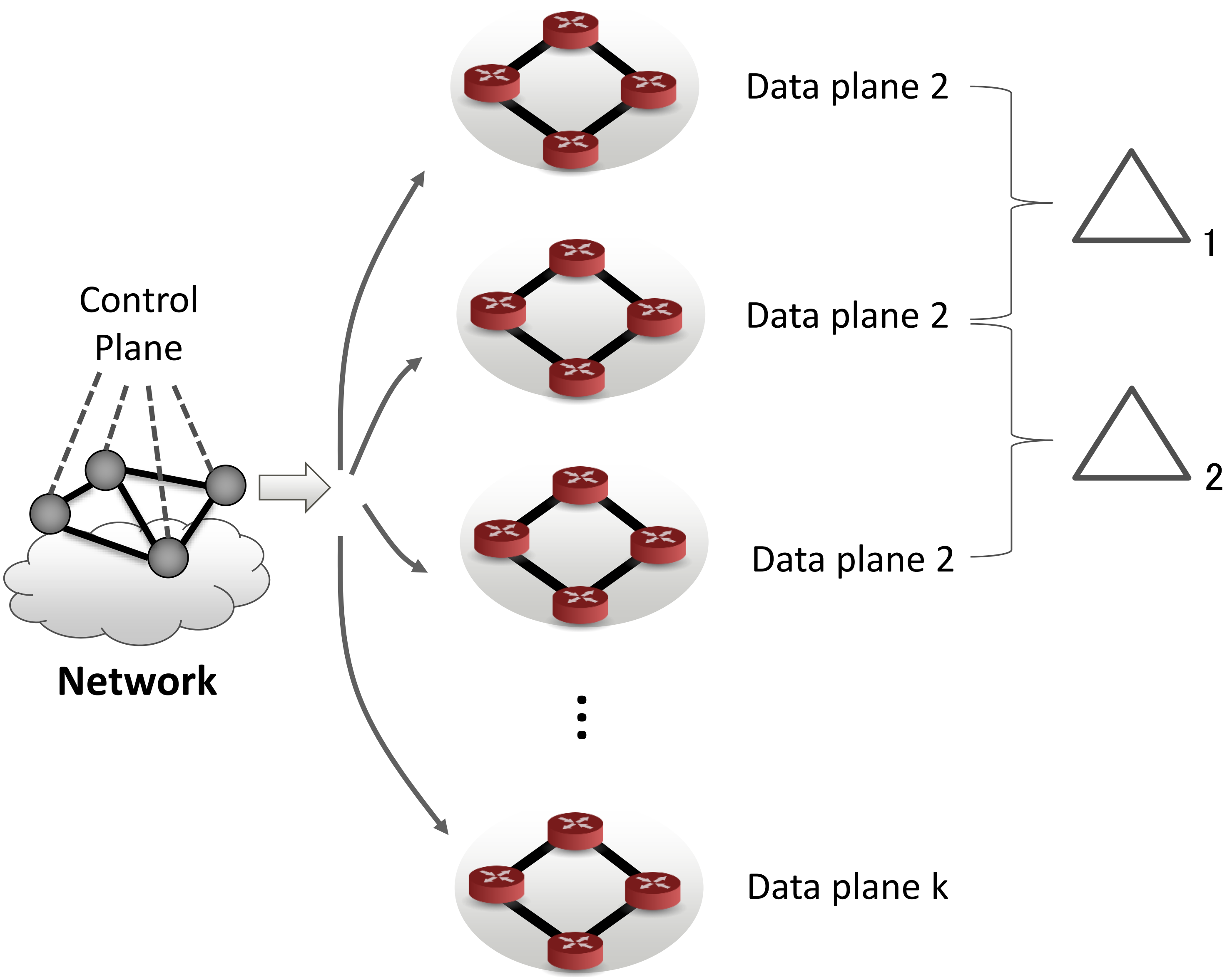
Taxonomy



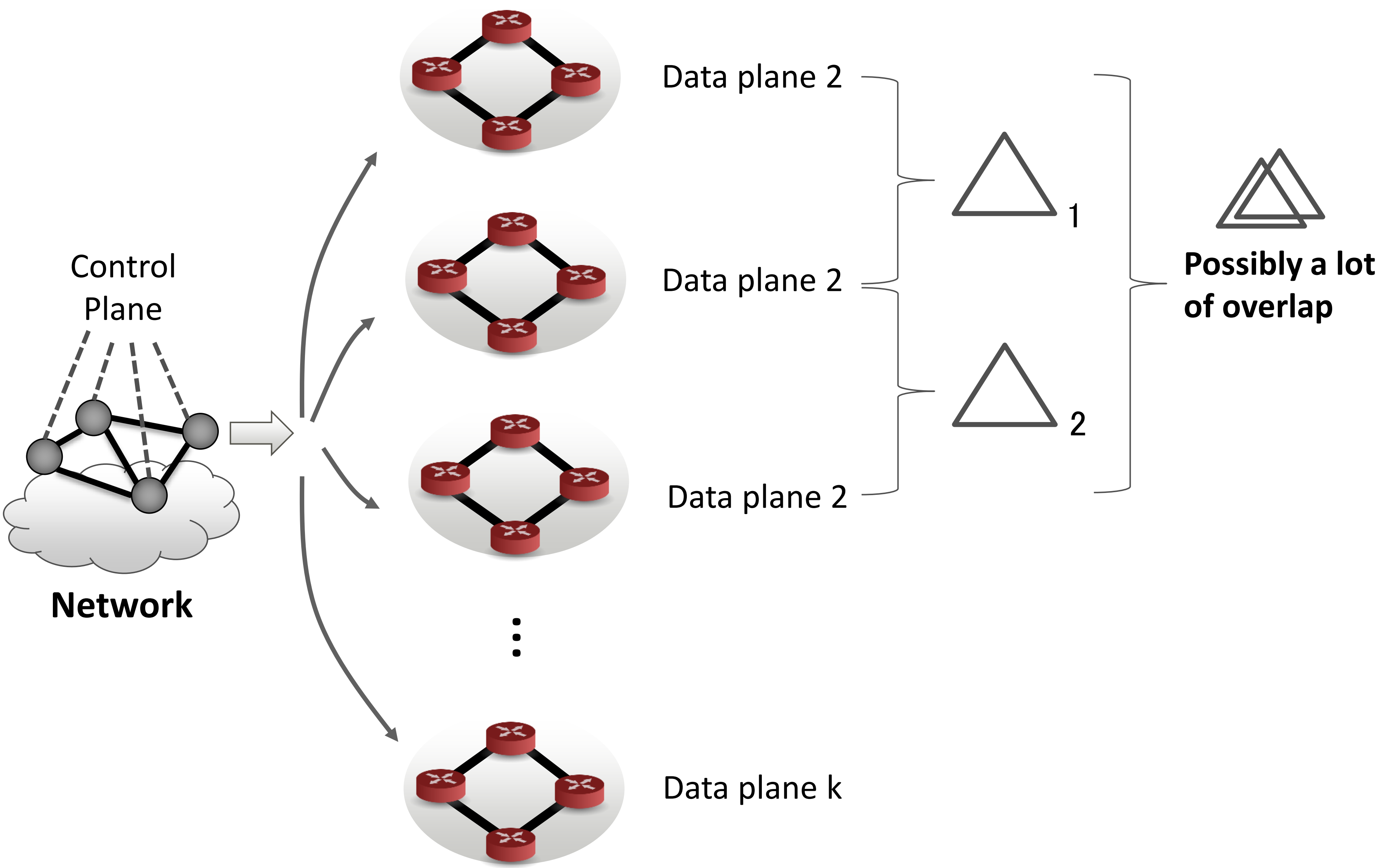
[Khurshid et al., NSDI 2013]
[Kazemian et al., NSDI 2013]
[Yang and Lam, ICNP 2013]

[Khurshid et al., NSDI 2013]

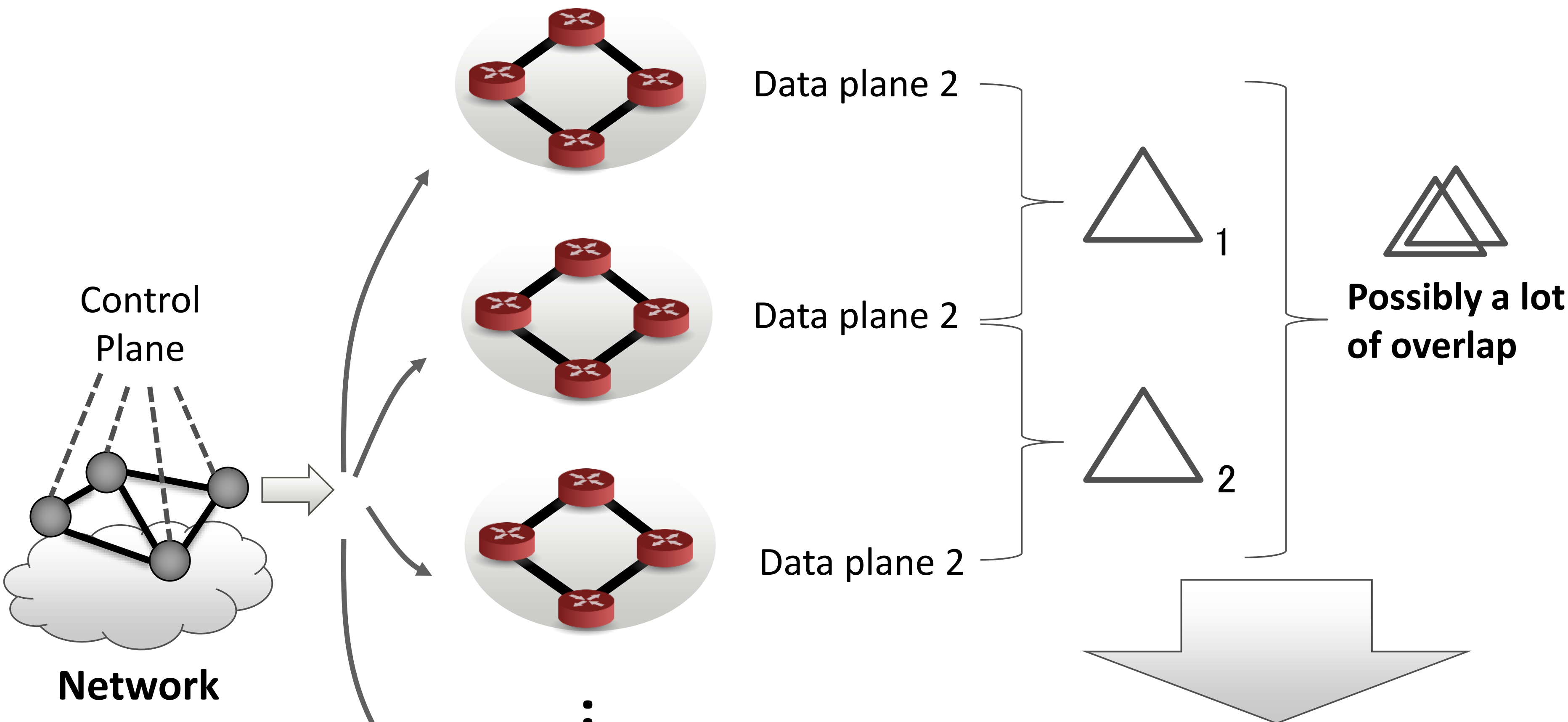
Incremental Network Verification

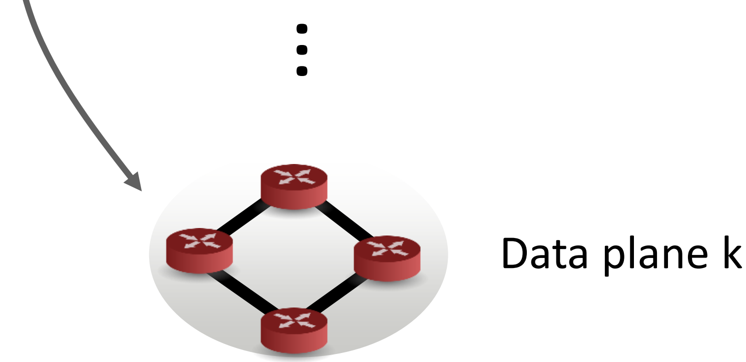


Incremental Network Verification



Incremental Network Verification





Problem: disruptive changes (e.g. outages) are still challenging to analyze in real-time.

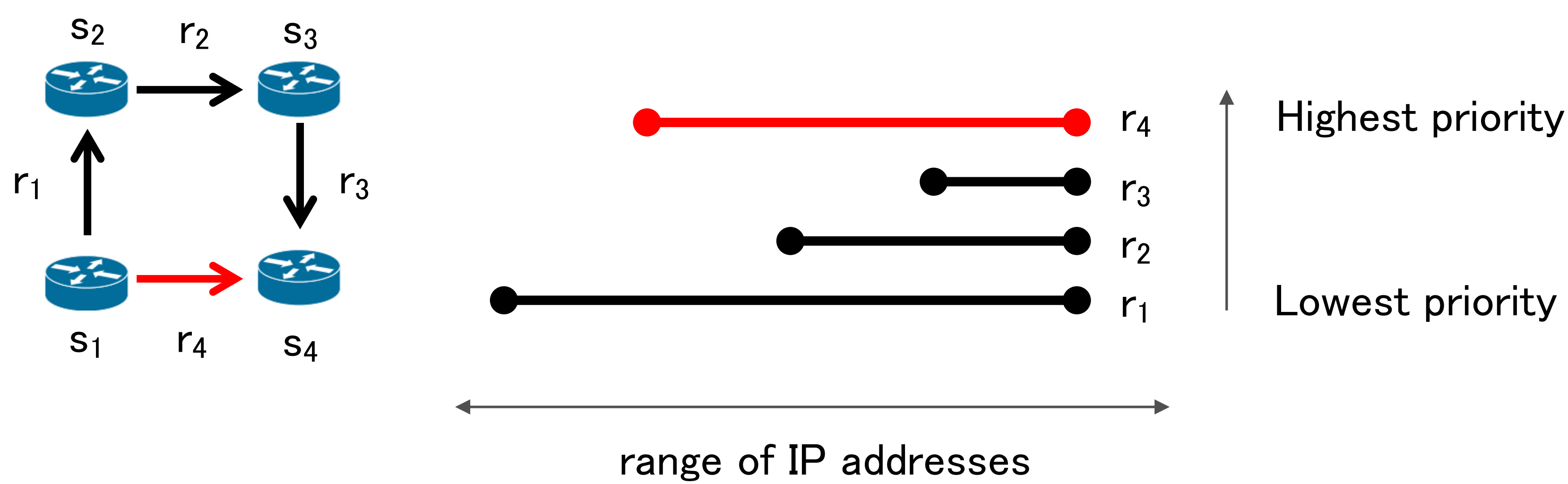
Our Contribution: Delta-net

“delta of deltas”

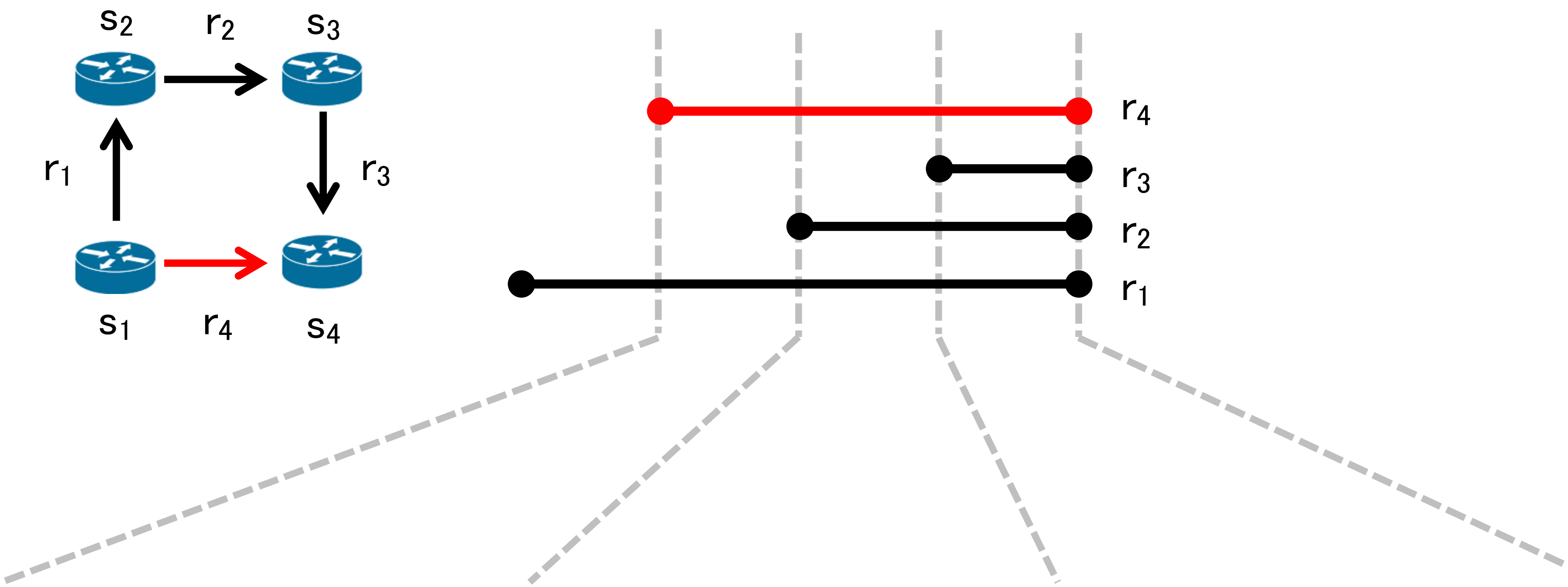
- Exploit similarity among forwarding behavior of packets through *parts* of the network, rather than its entirety.
  - **Experiments:** 10-100x faster on *network-wide* use cases

Example next …

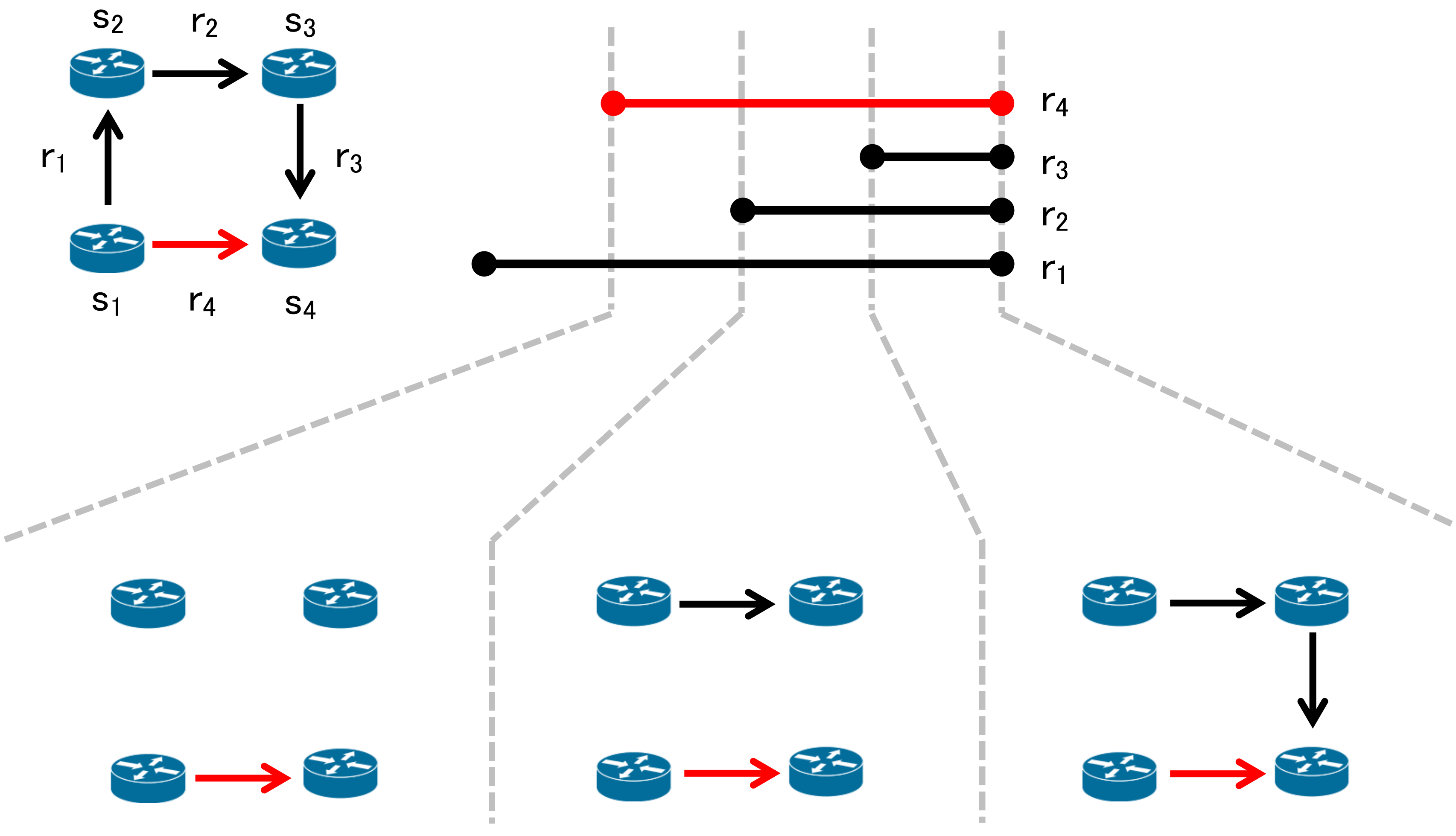
Forwarding Graphs Per Equivalence Class



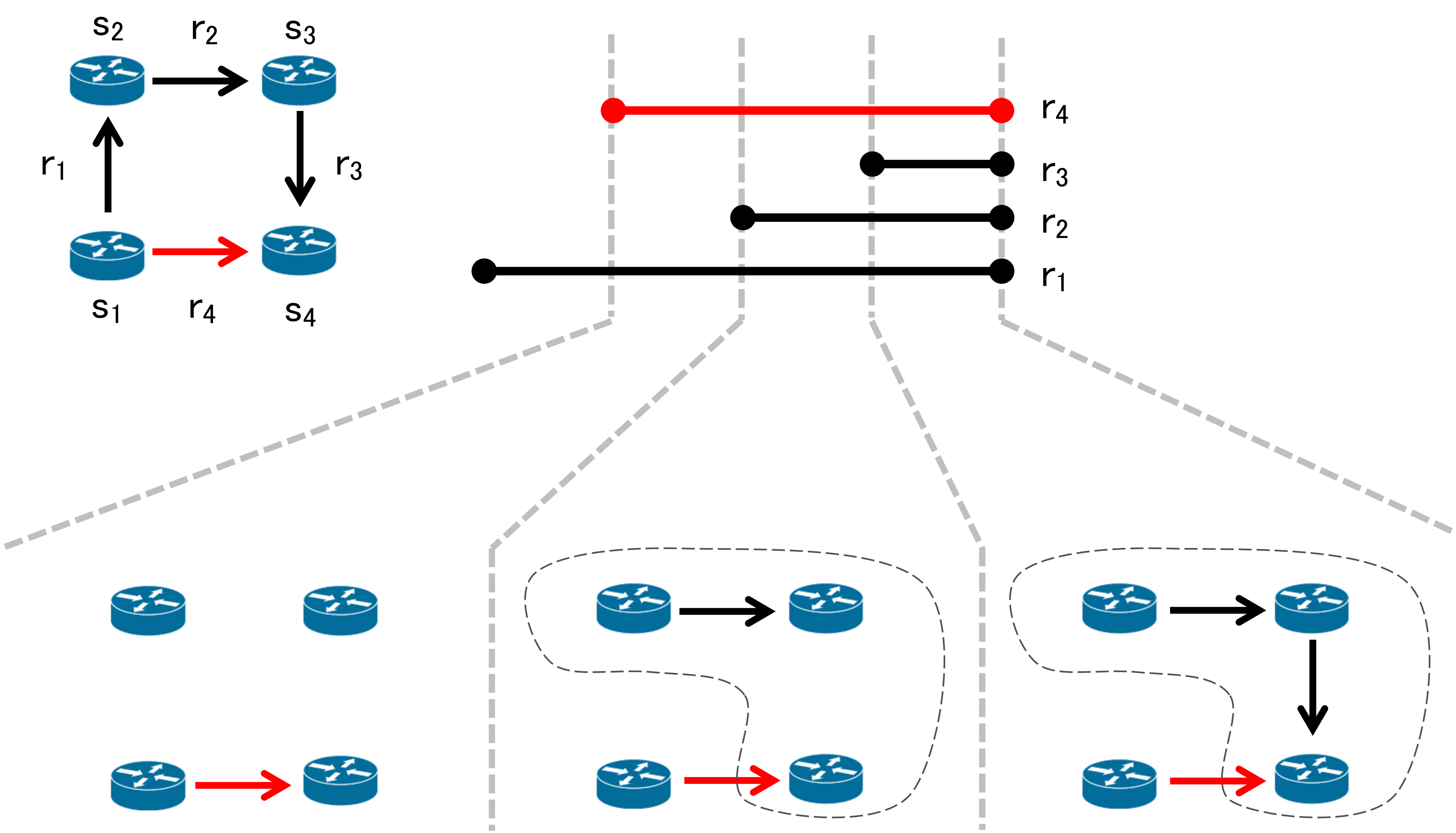
Forwarding Graphs Per Equivalence Class



Forwarding Graphs Per Equivalence Class

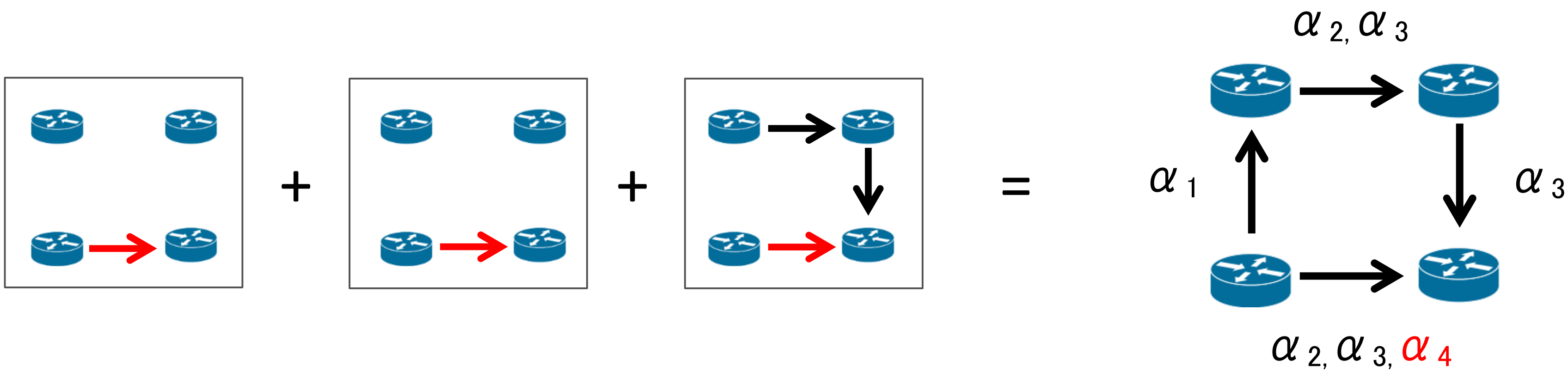


Forwarding Graphs Per Equivalence Class



Our Contribution: Delta-net

"delta of deltas"

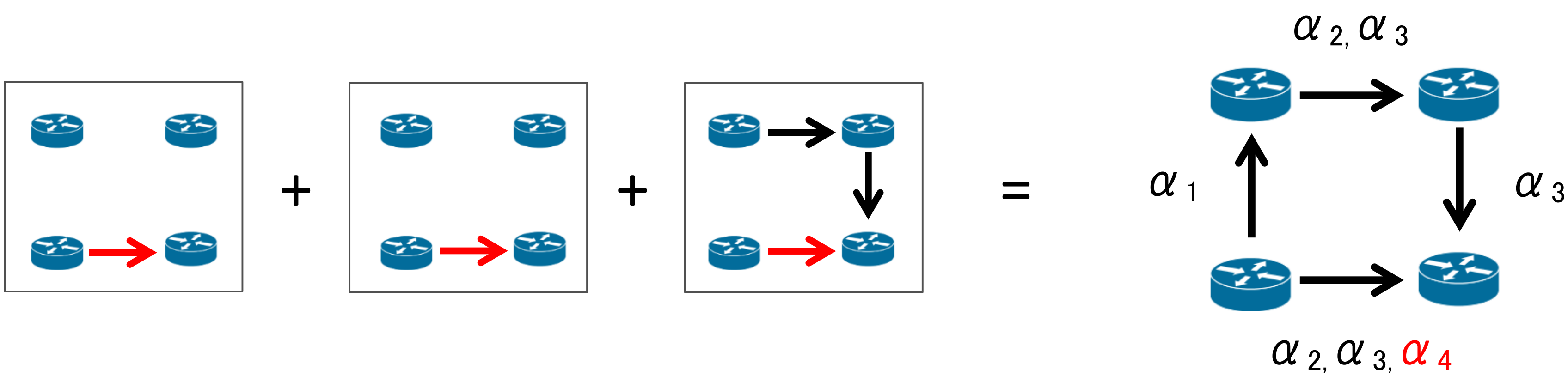


Rather than re-computing forwarding graphs …

incrementally maintain a single edge-labelled graph; represents *all* packet flows.

Our Contribution: Δ Delta-net

"delta of deltas"



Rather than re-computing forwarding graphs …

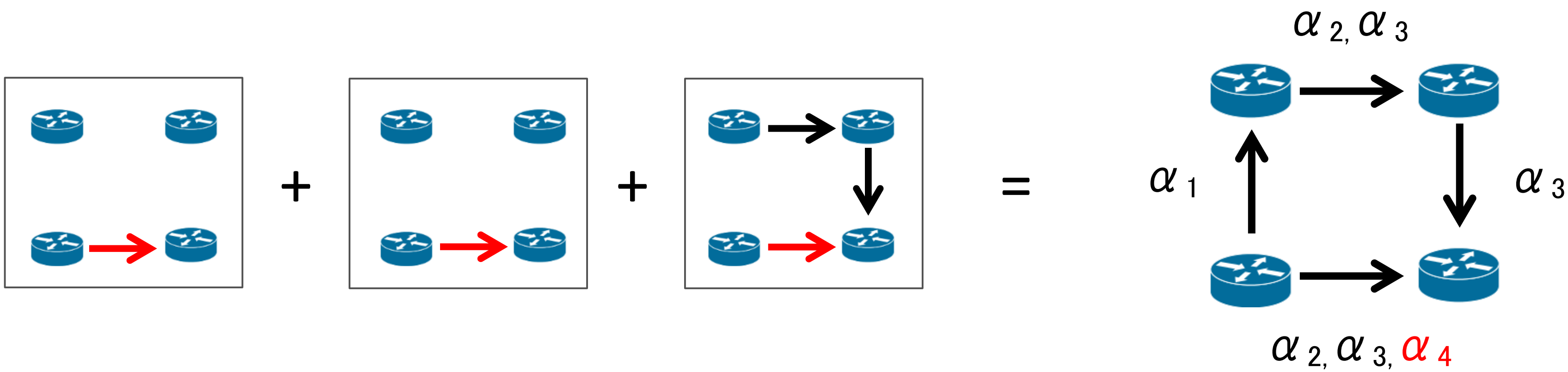
incrementally maintain a single edge-labelled graph; represents *all* packet flows.

- Single graph data structure to answer reachability queries, exposed through a simple C++ API.

Our Contribution: Delta-net

"delta of deltas"

atoms



Rather than re-computing forwarding graphs …

incrementally maintain a single edge-labelled graph; represents *all* packet flows.

- Single graph data structure to answer reachability queries, exposed through a simple C++ API.

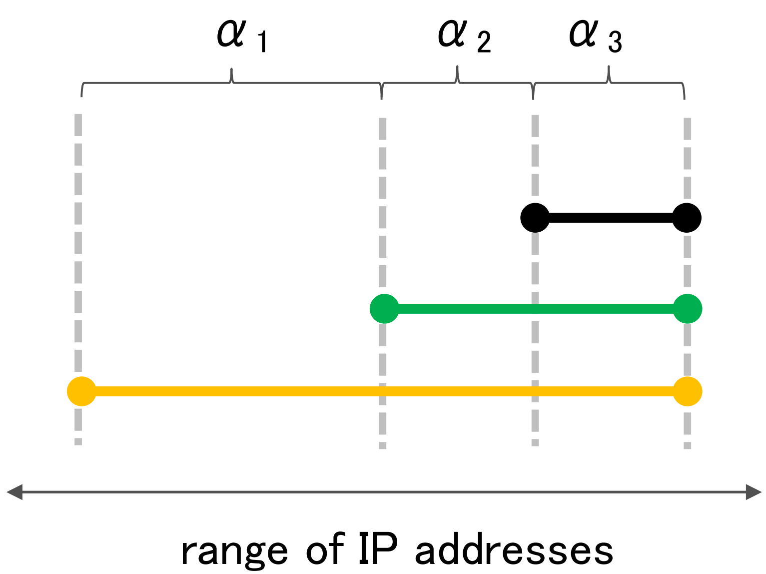
Atoms

$$\text{"Complex Stuff} = \sum \text{Simpler Stuff"}$$

Example: factorization into prime numbers, e.g.

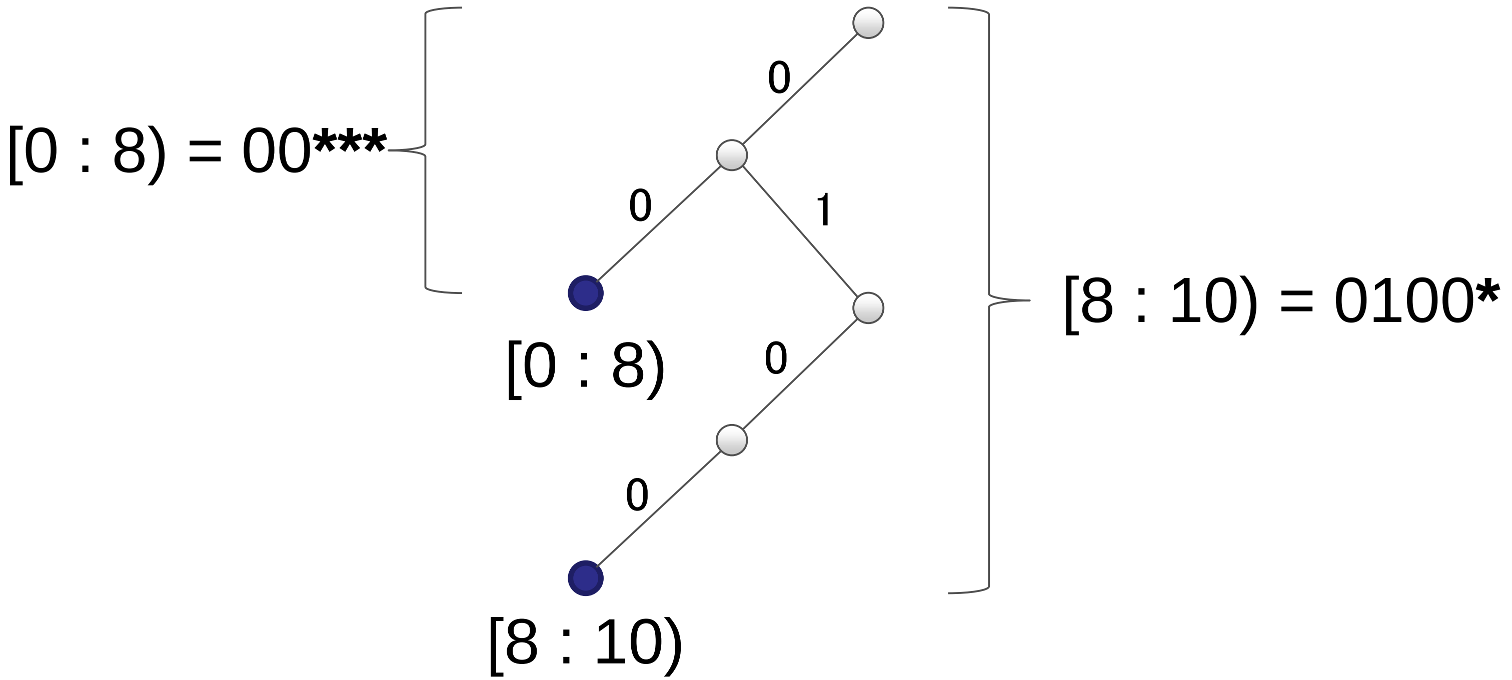
$$1479 = 3 \times 17 \times 29.$$

For IP prefix based networks, **atoms**:

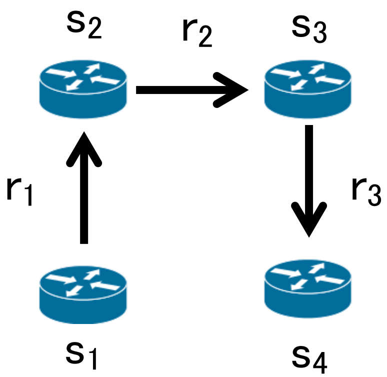


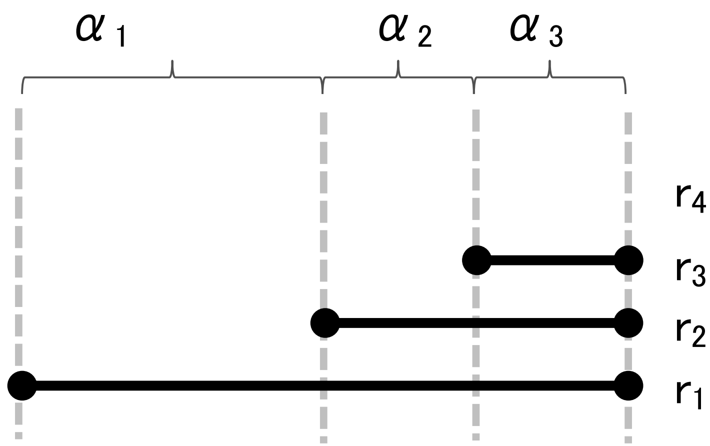
Compactness of Atoms

- More compact than a Patricia tree, e.g. consider α_0 = [0 : 10):

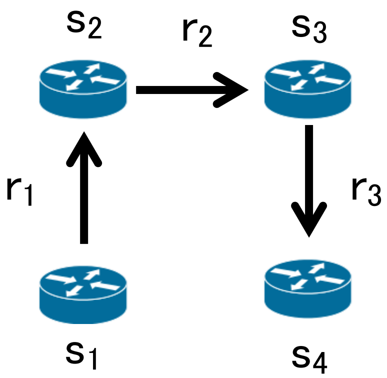


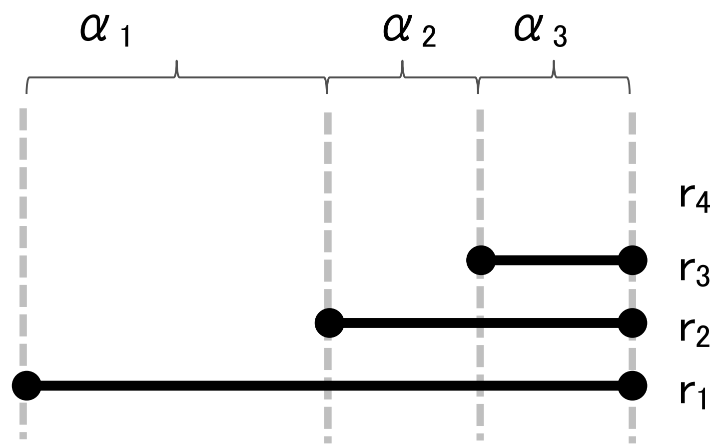
Atoms and Graph Transformation

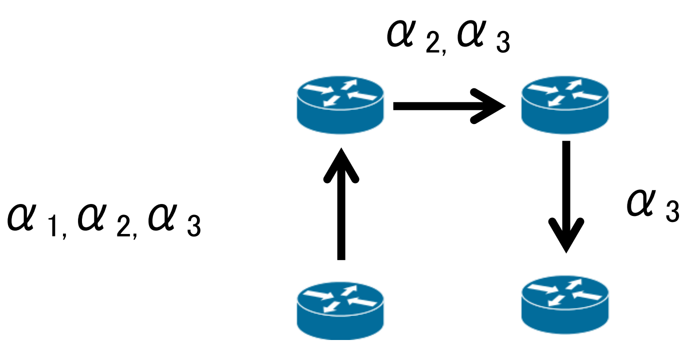




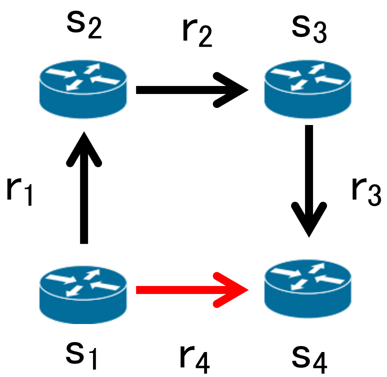
Atoms and Graph Transformation

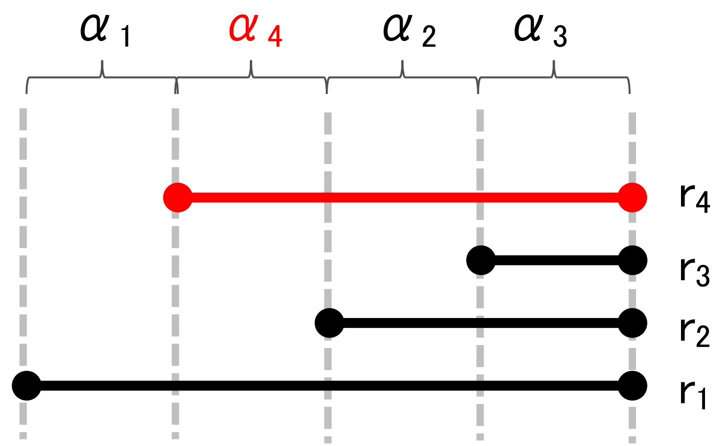


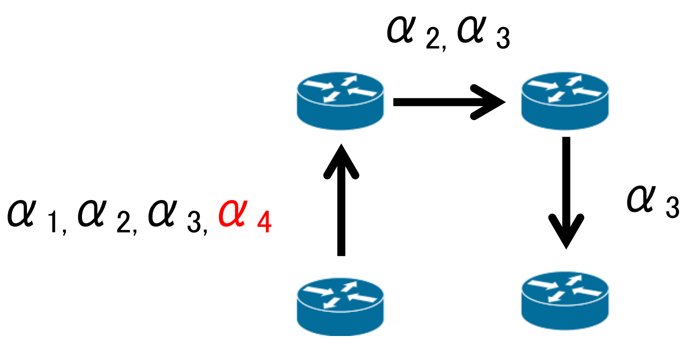




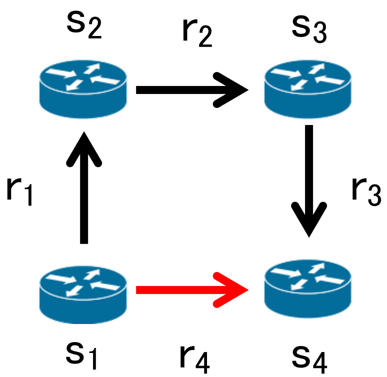
Atoms and Graph Transformation

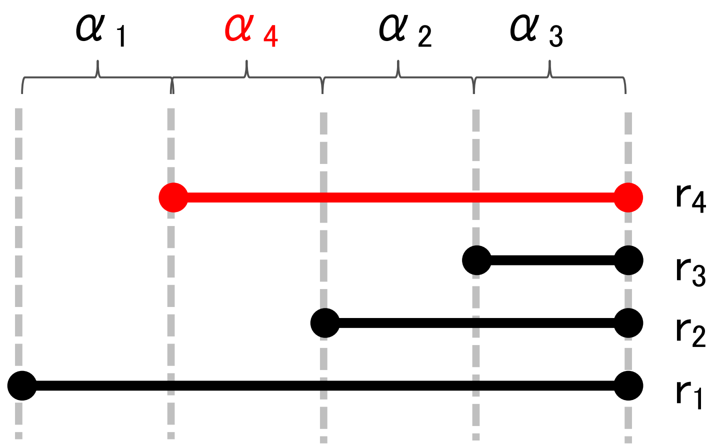


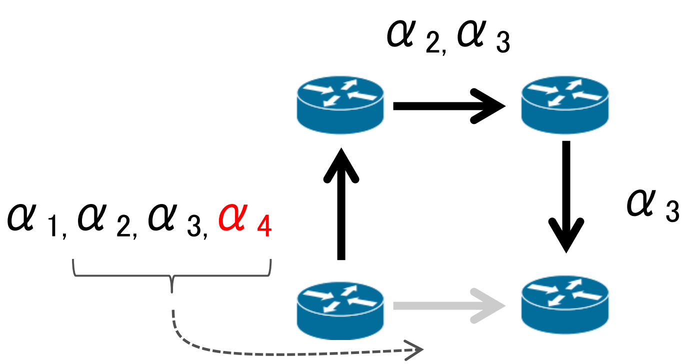




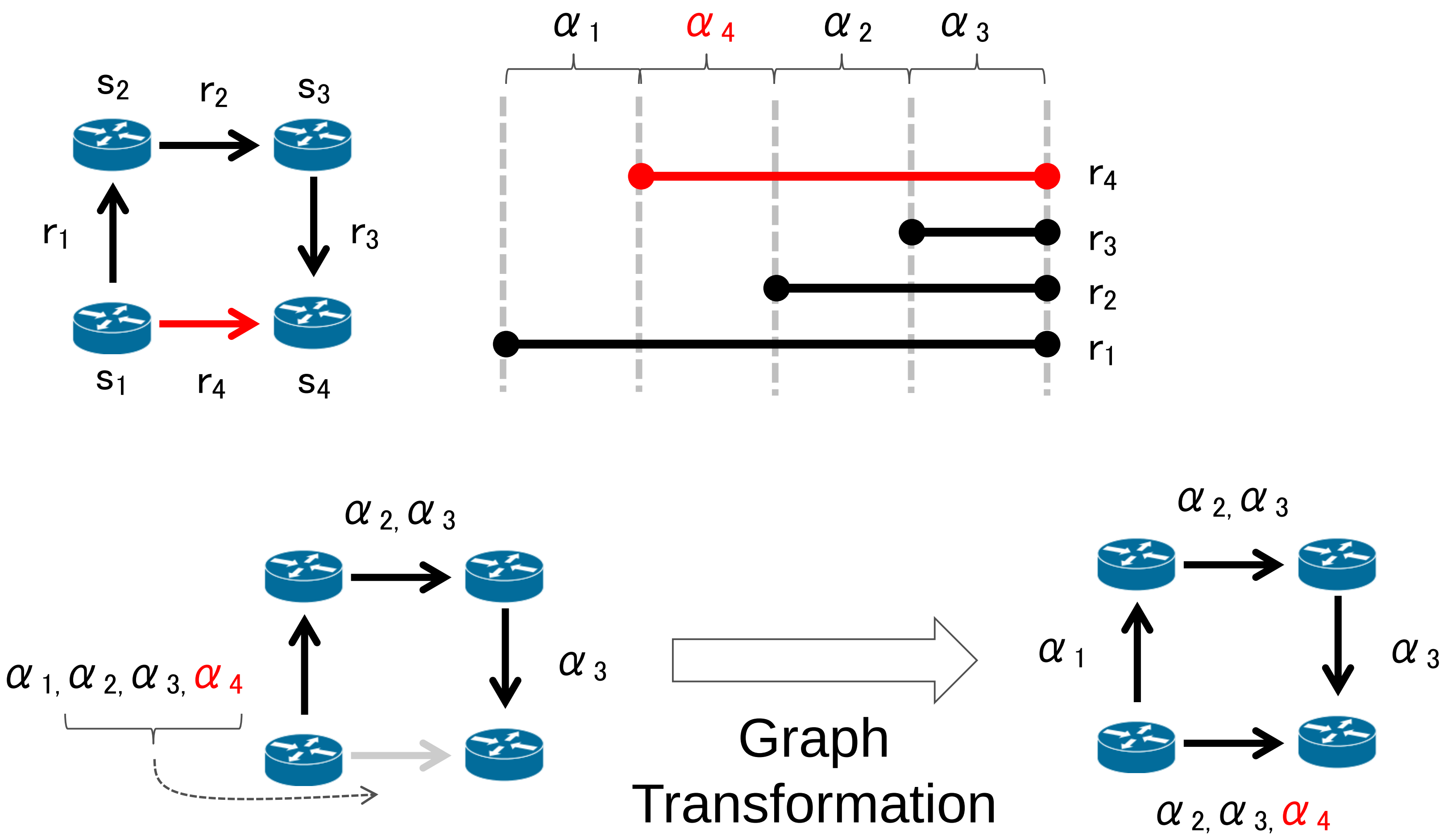
Atoms and Graph Transformation



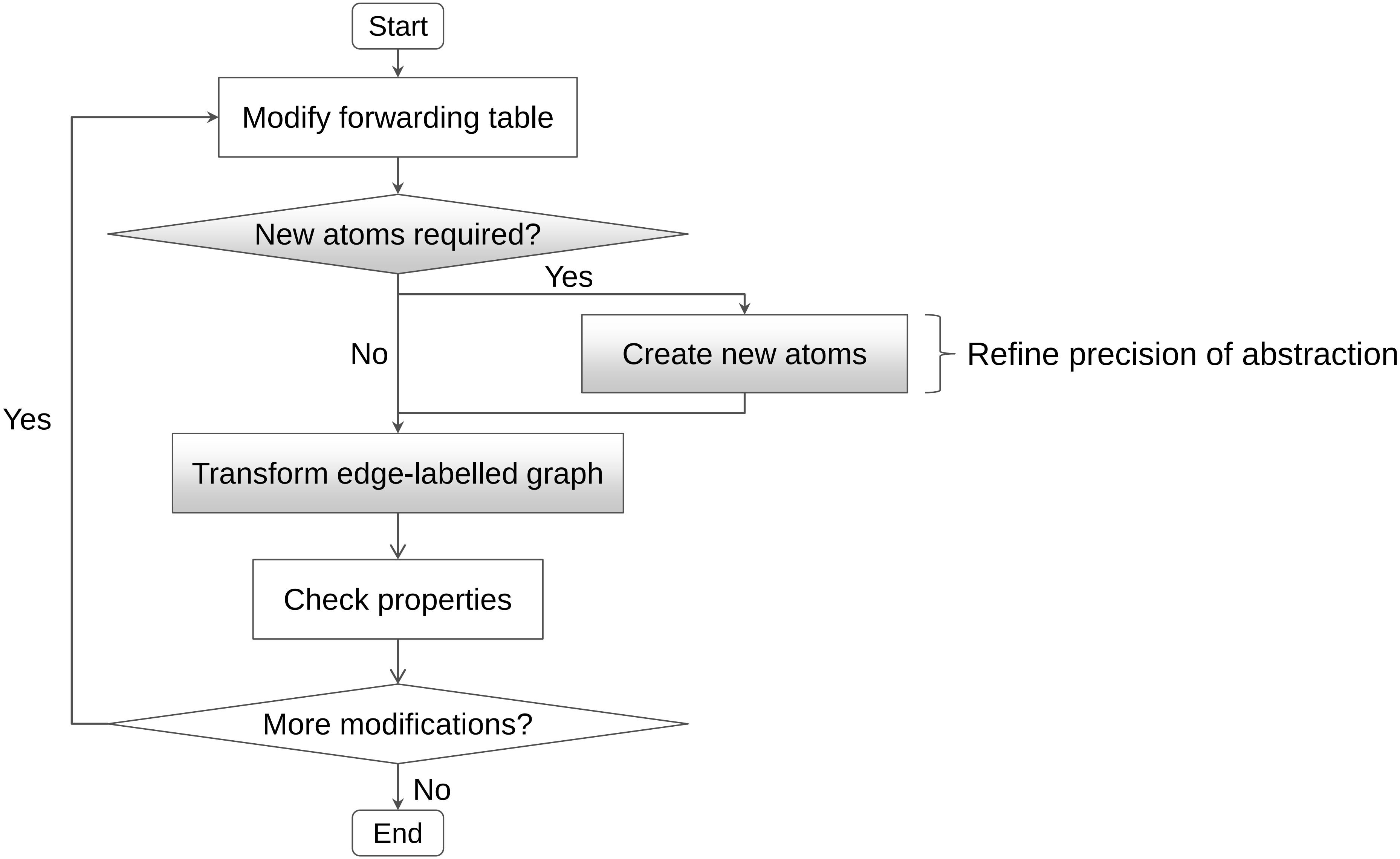




Atoms and Graph Transformation

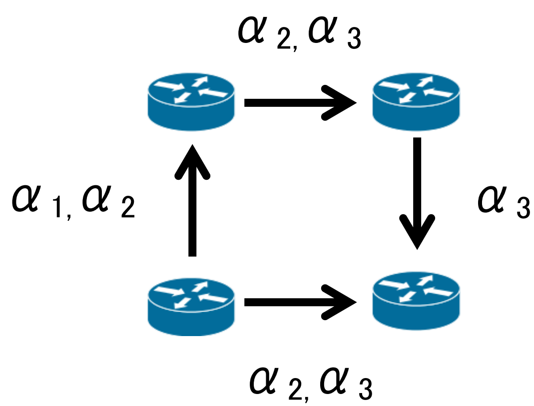


High-level Flowchart



All-Pairs Reachability

- Essential for Datalog-style “what-if” queries:

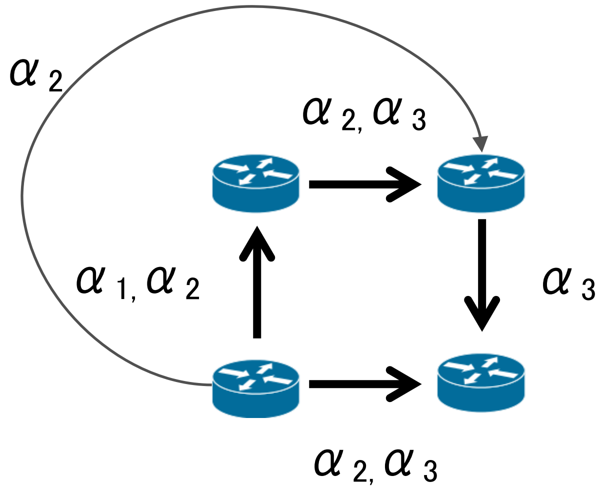


for k, i, j **in** V **do** ▷ Triple nested loop
    $label[i, j] \leftarrow label[i, j] \cup (label[i, k] \cap label[k, j])$
end for

Adaptation of Floyd–Warshall Algorithm

All-Pairs Reachability

- Essential for Datalog-style "what-if" queries:

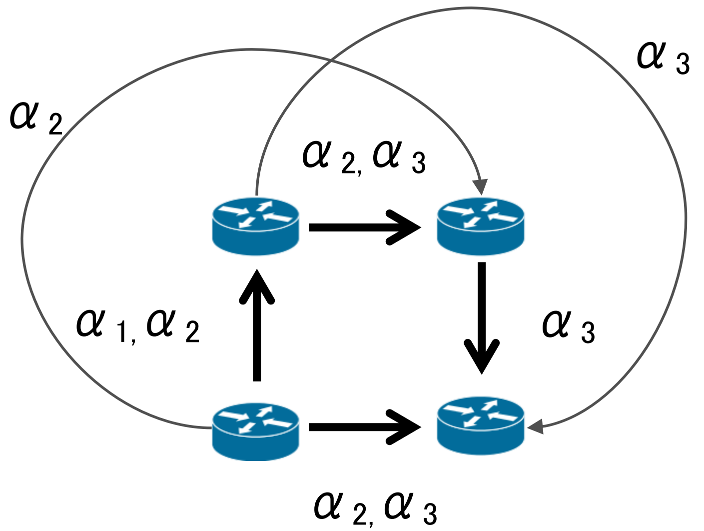


1: **for** k, i, j **in** V **do** ▷ Triple nested loop
2: $\quad label[i,j] \leftarrow label[i,j] \cup (label[i,k] \cap label[k,j])$
3: **end for**

Adaptation of Floyd–Warshall Algorithm

All-Pairs Reachability

- Essential for Datalog-style "what-if" queries:



1: **for** k, i, j **in** V **do** ▷ Triple nested loop
2: $label[i,j] \leftarrow label[i,j] \cup (label[i,k] \cap label[k,j])$
3: **end for**

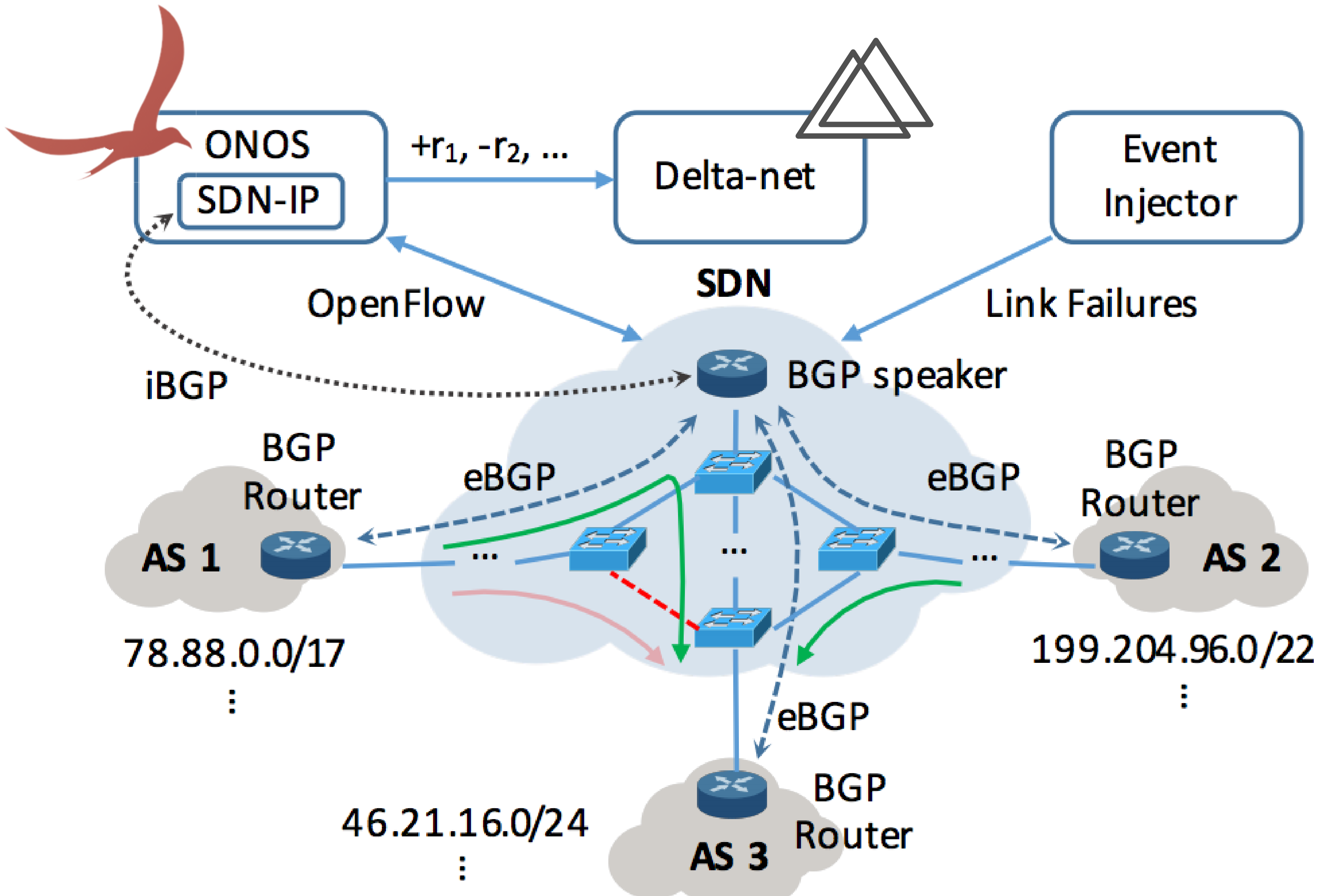
Adaptation of Floyd–Warshall Algorithm

Experimental Setup (2 Classes of Data Sets)

- Synthetic data sets similar to [Zeng et al., NSDI 2014]
  - Rocketfuel and Berkeley topologies from [Narayana et al., NSDI 2016]
  - IP prefixes from RouteViews project

- SDN-IP [Lin et al., SIGCOMM 2013] in ONOS
  - Globally deployed, ONOS flagship application

SDN-IP Experimental Setup



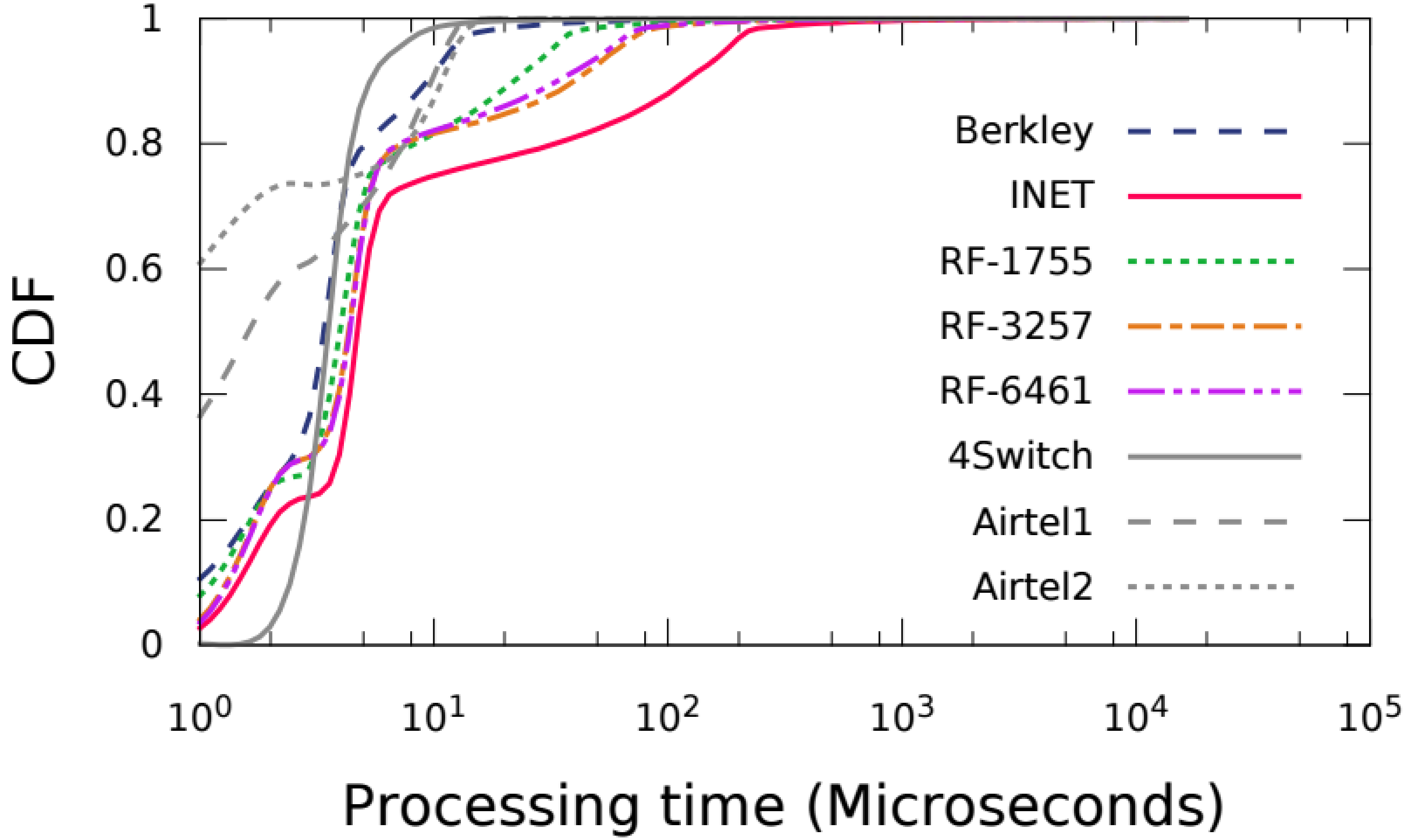
Data Sets

- Hundreds of million IP prefix rule insertions + removals

Data set	Nodes	Max Links	Operations	
Berkeley	23	252	25.6×10^6	
INET	316	$40,770$	249.5×10^6	
RF 1755	87	$2,308$	67.5×10^6	Synthetic
RF 3257	161	$9,432$	149.0×10^6	
RF 6461	138	$8,140$	150.0×10^6	
Airtel 1	68	260	14.2×10^6	
Airtel 2	68	260	505.2×10^6	SDN–IP
4Switch	12	16	1.12×10^6	

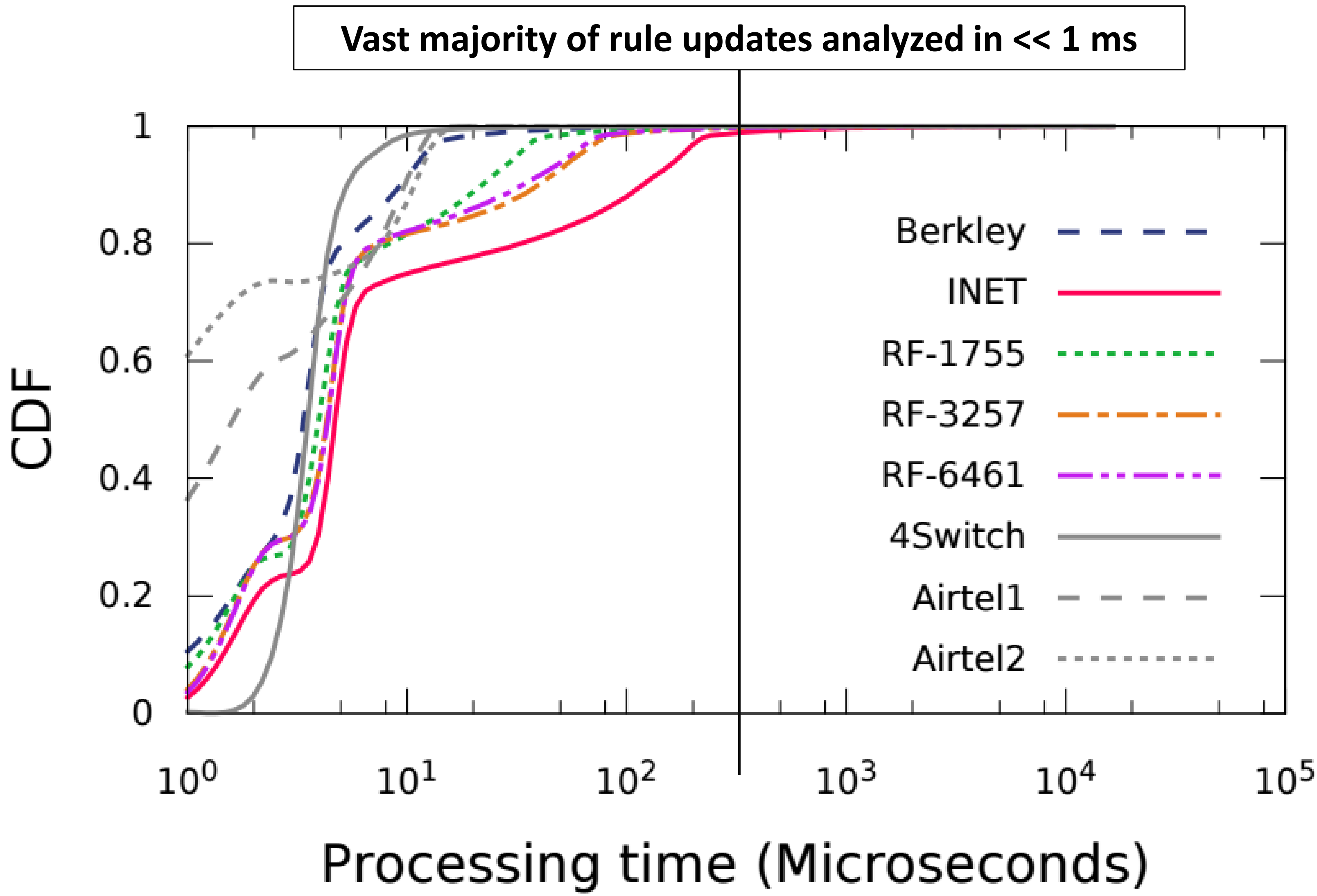
Experiments: Measuring Rule Updates

- Find all forwarding loops introduced by a rule insertion.



Experiments: Measuring Rule Updates

- Find all forwarding loops introduced by a rule insertion.



Experiments: Beyond Network Updates

- What parts of the network are affected by link failures?
  - Query proposed by [Khurshid et al., NSDI 2013].

Data plane	**Rules**	**Average query time** (ms)	
		Veriflow-RI	Delta-net
Berkeley	12,817,902	3,073.0	**4.7**
INET	124,733,556	29,117.5†	**0.7**
RF 1755	33,732,869	8,100.6	**1.3**
RF 3257	74,492,920	17,645.3†	**1.0**
RF 6461	75,005,738	17,594.5†	**0.4**
Airtel	38,100	4.5	**0.04**
4Switch	1,120,000	433.4	**21.1**

  - **Summary:** Delta-net can answer queries where Veriflow-RI times out.

Concluding Remarks

- Delta-net, a new real-time data plane checker.
  - Our research considers the “delta of deltas”.
  - Opens up new Datalog-style use cases, previously out of reach.
  - Data sets publically available now:
    https://github.com/delta-net/datasets

For questions and comments, contact: ahorn@us.fujitsu.com.

We are also looking for industry/academic partners, interns etc.

- Future work:
  - Parallelization
  - Multi-range support
  - Avoid space/time trade-offs, at what cost for query expressiveness?