

# Differential Network Analysis

Peng Zhang, Aaron Gember-Jacobson,  
Yueshang Zuo, Yuhao Huang, Xu Liu, Hao Li





**Change is the  
only constant**

~ Heraclitus



Configuration  
edits

 **WISCONSIN**  
UNIVERSITY OF WISCONSIN-MADISON  
 **Georgia  
Tech.**  
*55 stanzas/router/month*



*12 edits/  
router/week*

**Change is the  
only constant**



Configuration  
edits

 **WISCONSIN**  
UNIVERSITY OF WISCONSIN-MADISON  
 **Georgia  
Tech.**  
*55 stanzas/router/month*



*12 edits/  
router/week*



External  
route  
updates

 **AT&T**  **Level(3)**  
COMMUNICATIONS  **Sprint**  
*100K updates/day*

**Change is the  
only constant**



Configuration edits

 **WISCONSIN**  
UNIVERSITY OF WISCONSIN-MADISON  
 **Georgia Tech.**  
*55 stanzas/router/month*



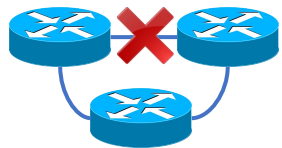
*12 edits/  
router/week*



External route updates

 **AT&T**  **Level(3)**  
COMMUNICATIONS  **Sprint**  
*100K updates/day*

**Change is the only constant**



Link/router failures

**CENIC**  
*38 failures/link/year*

 **Microsoft**  
*18 link failures/day*



**Change causes  
outages**

# Facebook outage: what went wrong and why did it take so long to fix after social platform went down?

**Josh Taylor**

🐦 @joshgnosis

Tue 5 Oct 2021 01:53 EDT

Facebook issued [a statement](#) on Tuesday confirming that the cause of the outage was a **configuration change** to the backbone routers that coordinate network traffic between the company's data centres, which had a cascading effect, bringing all Facebook services to a halt.

# Change causes outages

## Google Accidentally Broke Japan's Internet

One mistake from a Google engineer meant hours without internet access for much of Japan.

// BY [AVERY THOMPSON](#) AUG 28, 2017

The problem started when Google **updated an internet routing table**, which is essentially a list of IP addresses with instructions on how to get to them. Google was trying to tell the world which web addresses it owned, but accidentally included several addresses that belonged to some Japanese telecoms.



The New York Times

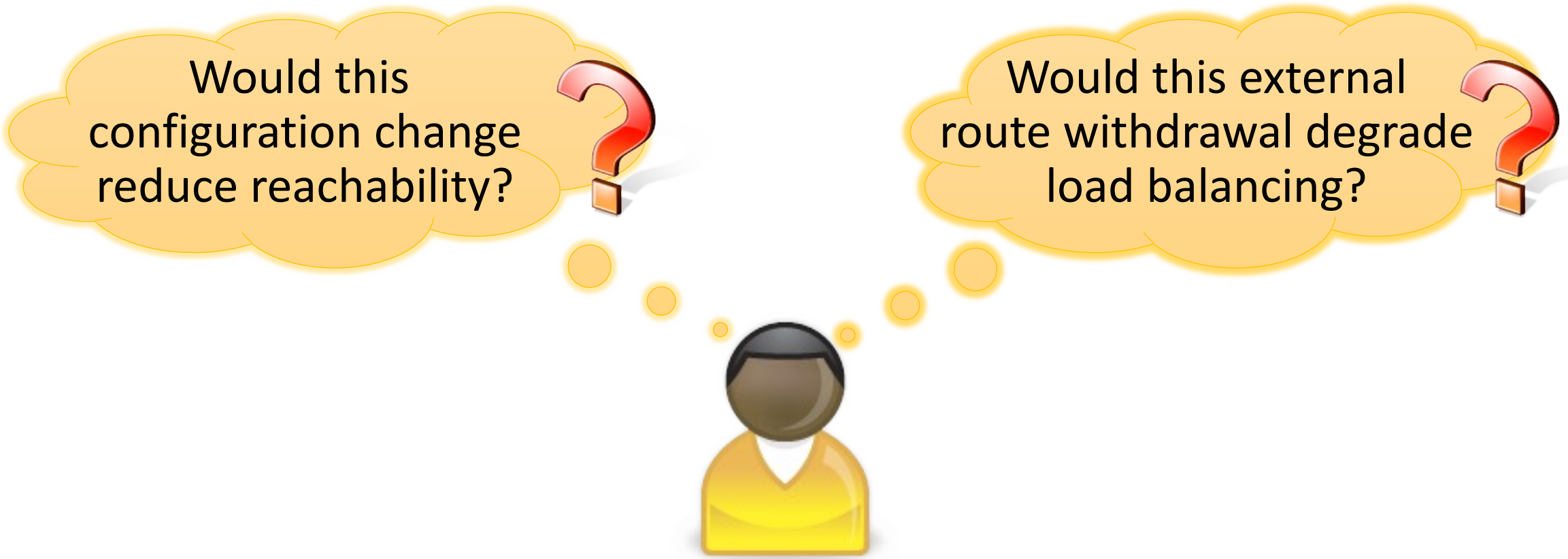


## *United Airlines Grounds Flights, Citing Computer Problems*

By [Christopher Drew](#)

July 8, 2015

[United Airlines](#) grounded planes nationwide for nearly two hours Wednesday morning after a **faulty computer network router** disrupted its passenger reservations system.

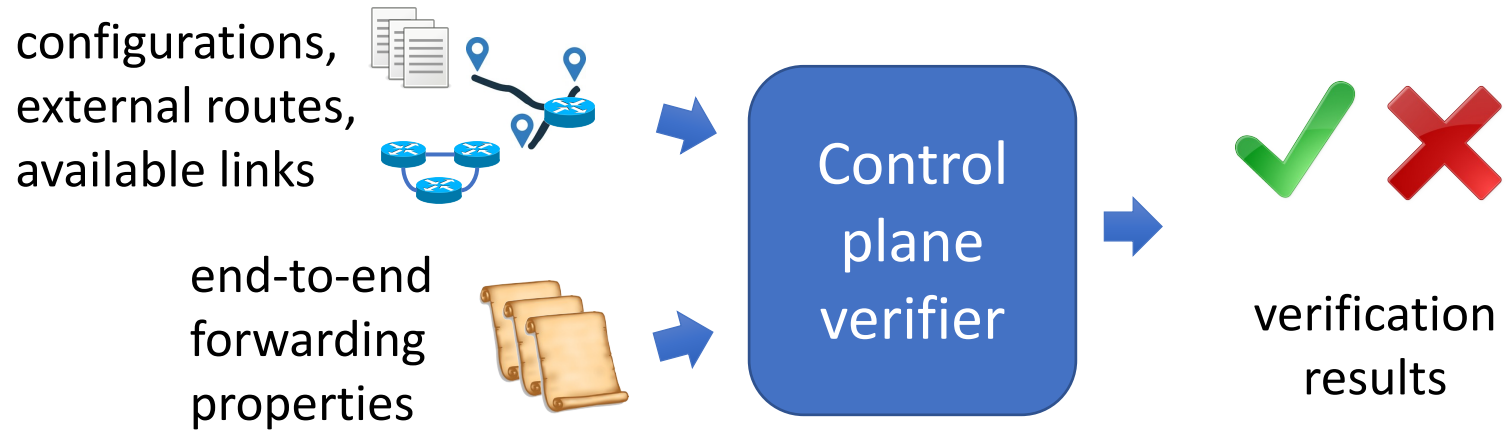


Would this configuration change reduce reachability?

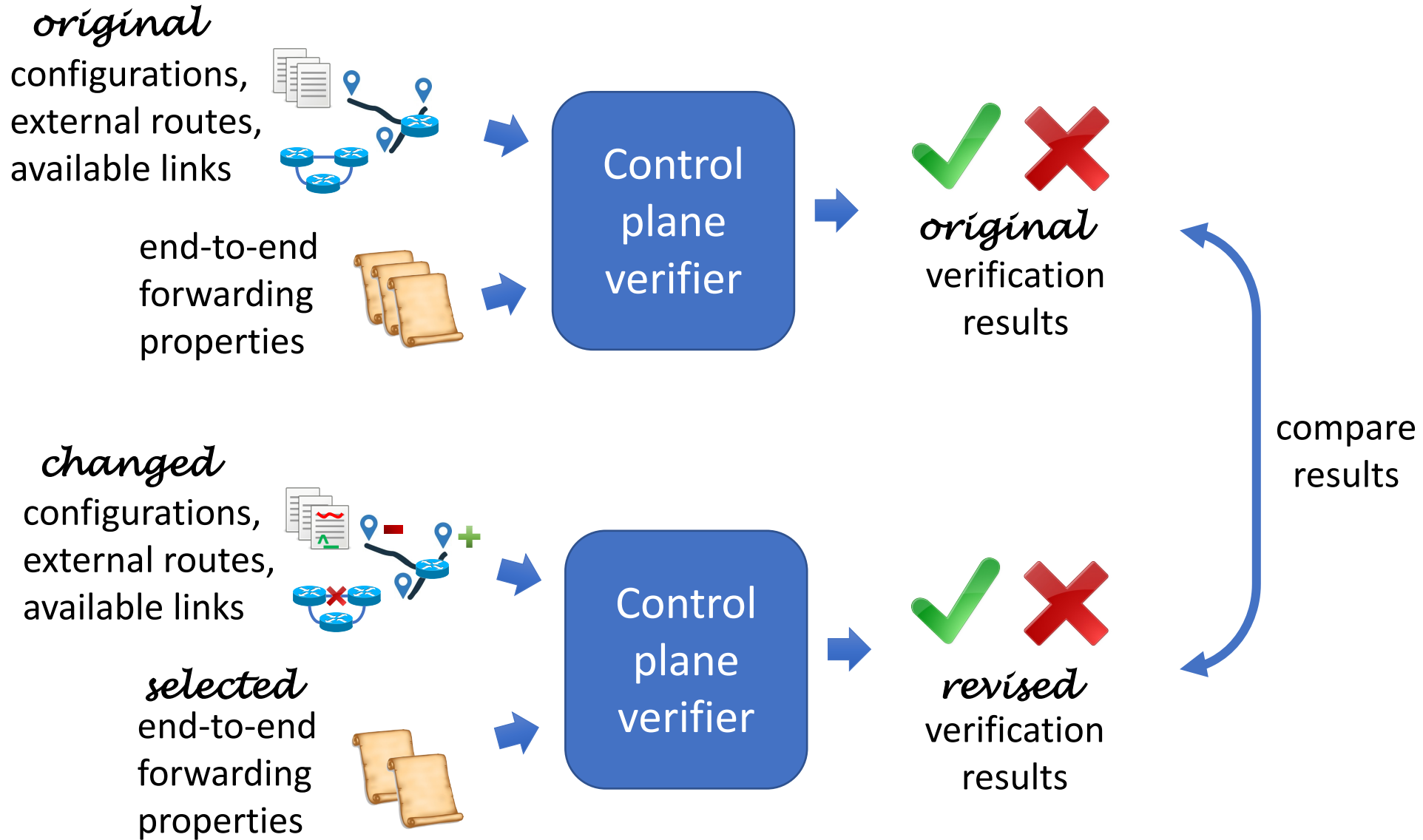
Would this external route withdrawal degrade load balancing?



# Existing control plane verifiers



# Existing control plane verifiers



# Changes are often small ↴ Analyzing from scratch is wasteful



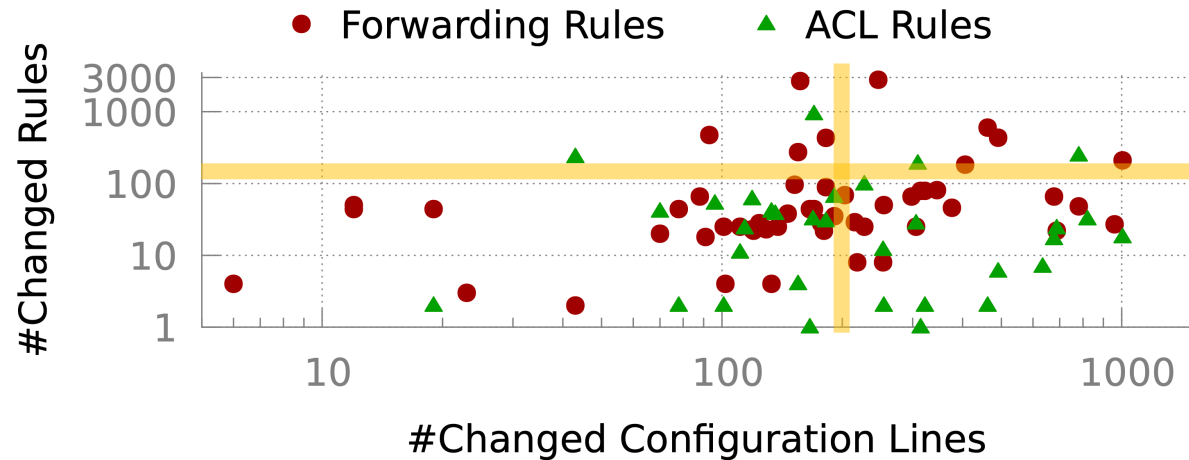
Analyze 3 months of  
configuration edits



28 routers

~75K lines of configuration

~25K forwarding rules



# Changes are often small ↴ Analyzing from scratch is wasteful



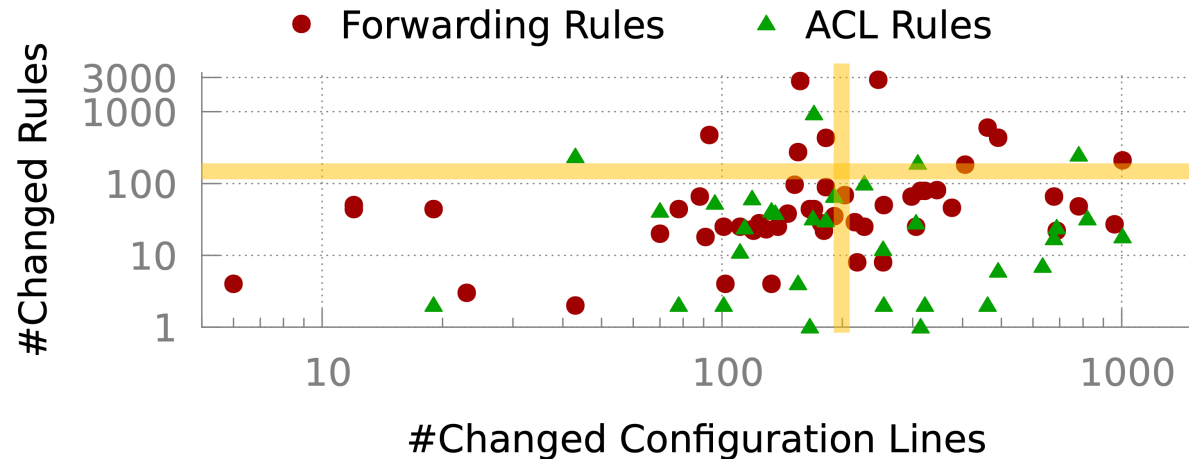
Analyze 3 months of  
configuration edits



28 routers

~75K lines of configuration

~25K forwarding rules



Analyze 1 year of  
hourly RIB snapshots

**INTERNET**<sup>®</sup>

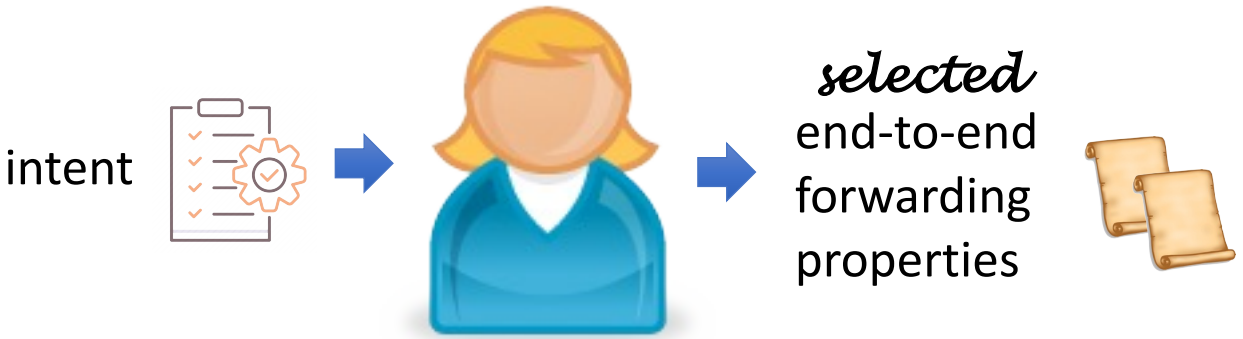
85% of hours:

No RIB change

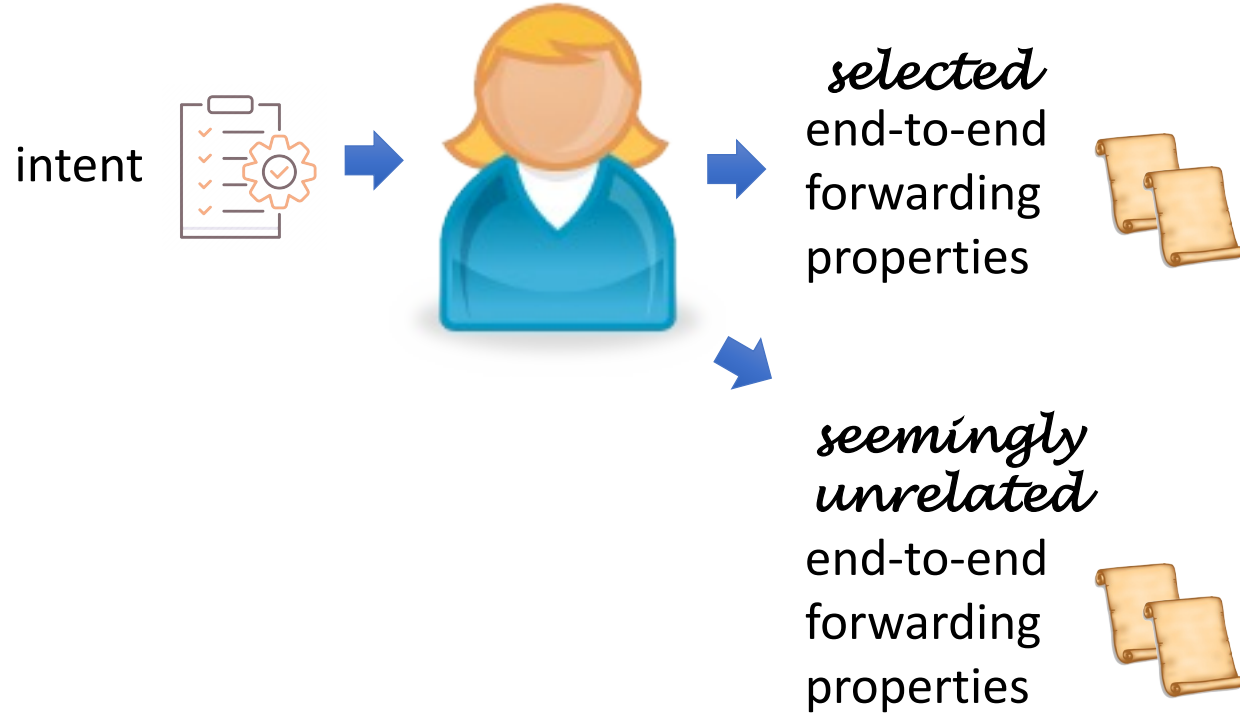
4% of hours:

10 RIB changes

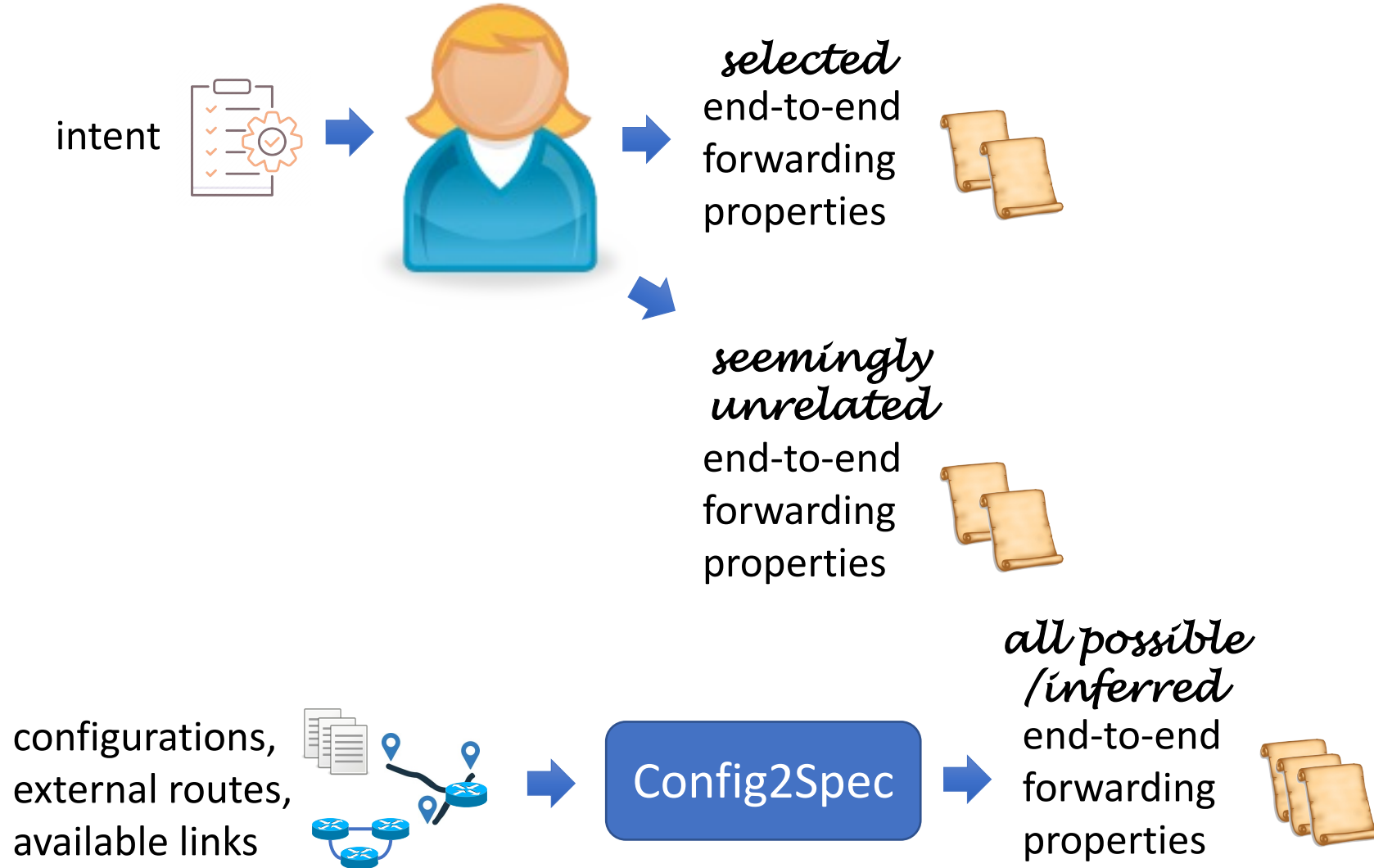
# Hard to select properties to reverify



# Hard to select properties to reverify



# Hard to select properties to reverify



*Differences*  
should be first  
class citizens!

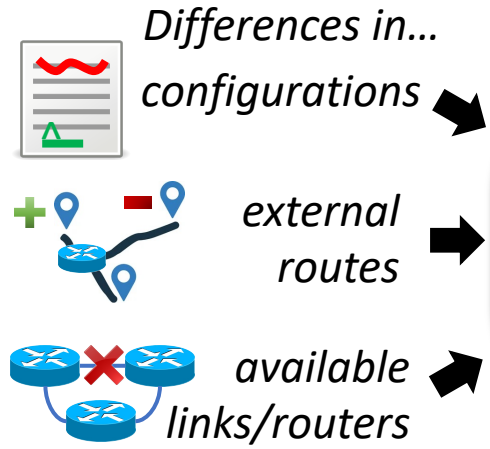




# New paradigm

Differential **N**etwork **A**nalysis

# New paradigm



Differential Network Analysis

# New paradigm

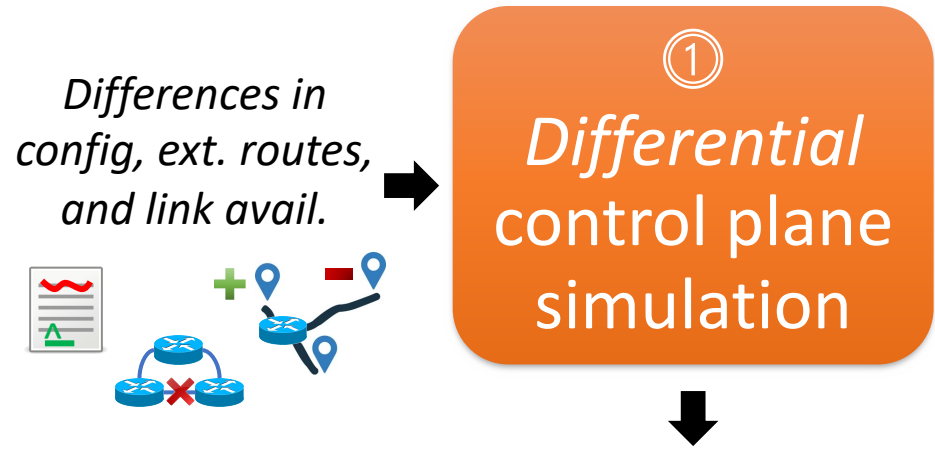


# New paradigm



How to make DNA efficient/  
easy to extend ?

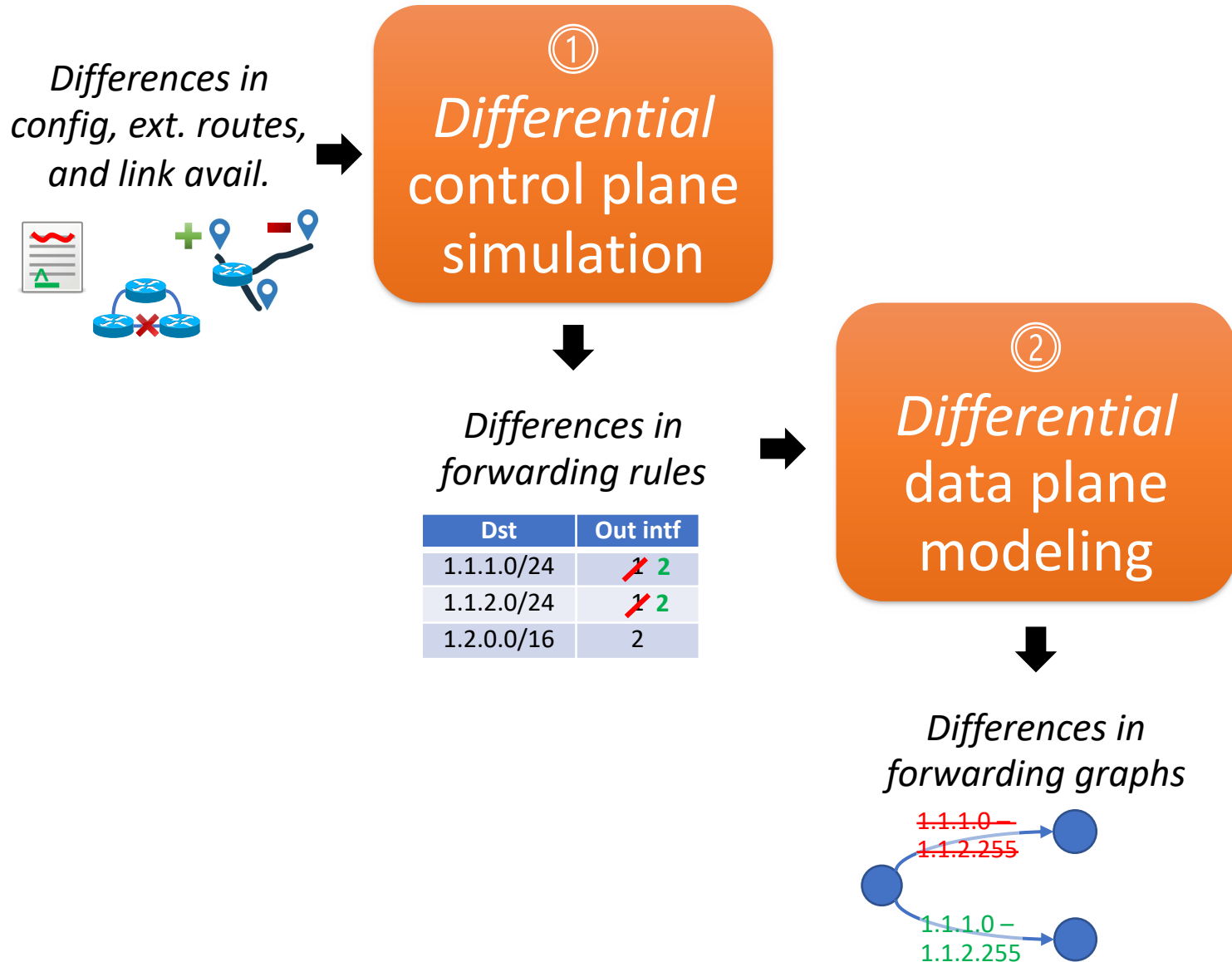
# Modular, three-stage framework



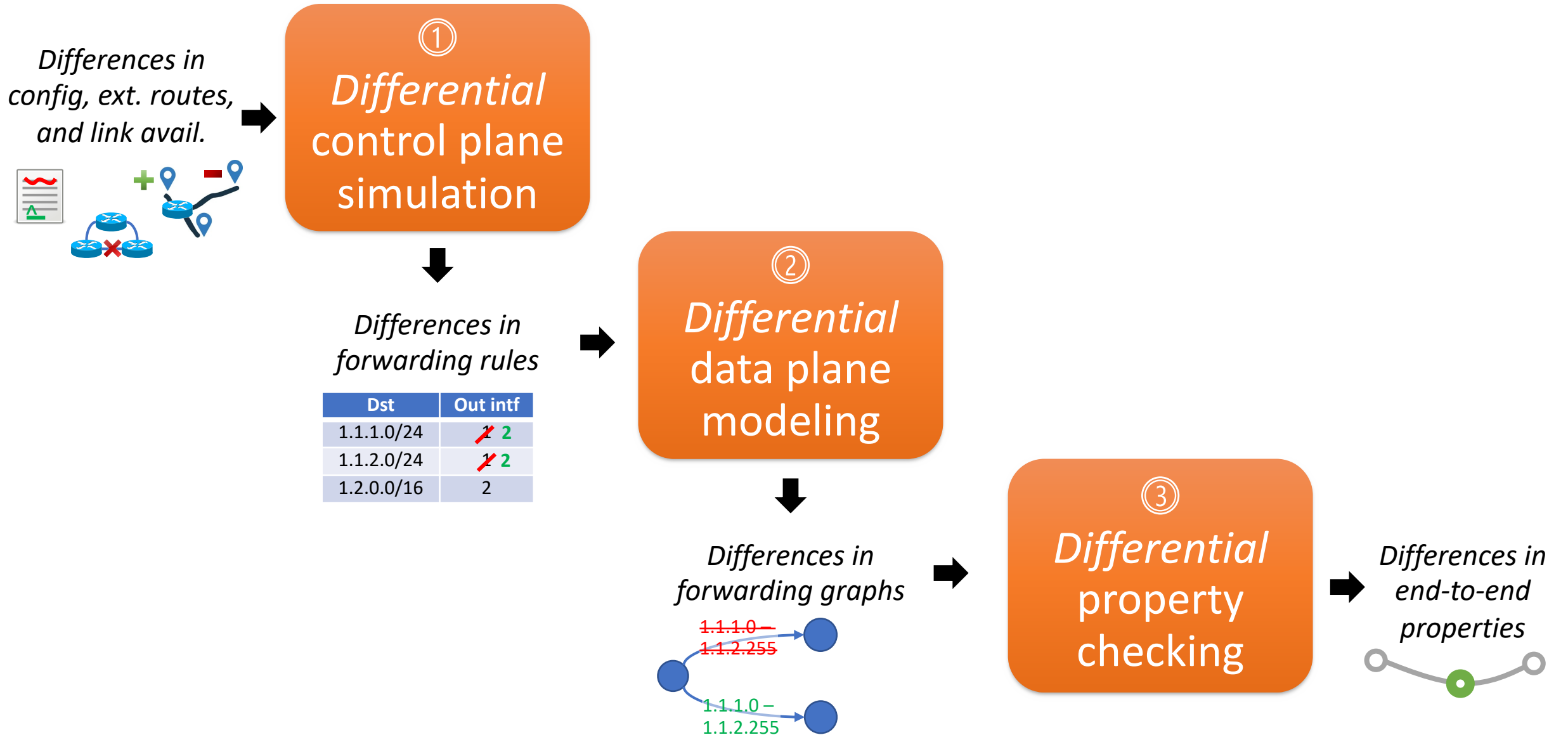
*Differences in forwarding rules*

Dst	Out intf
1.1.1.0/24	<del>1</del> 2
1.1.2.0/24	<del>1</del> 2
1.2.0.0/16	2

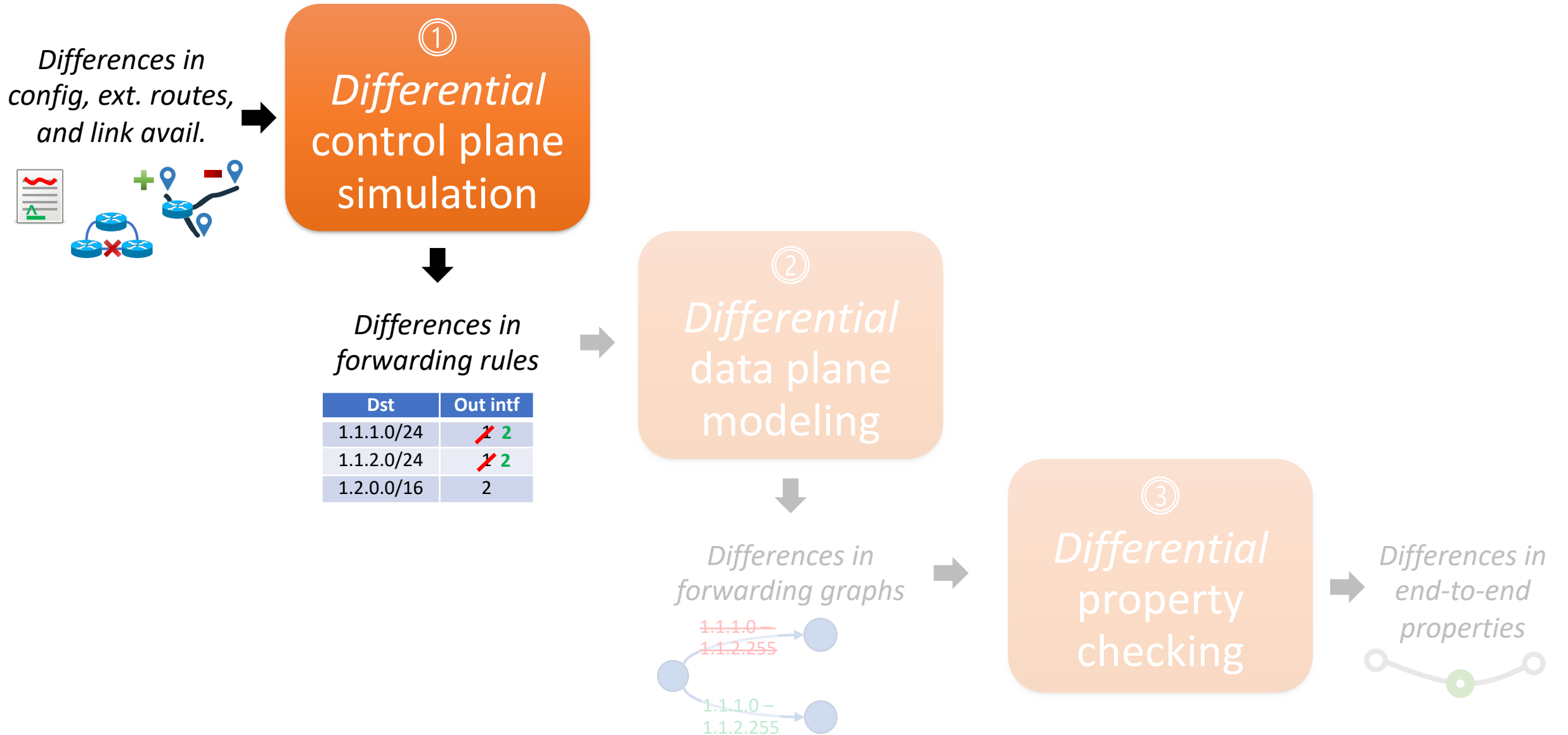
# Modular, three-stage framework



# Modular, three-stage framework



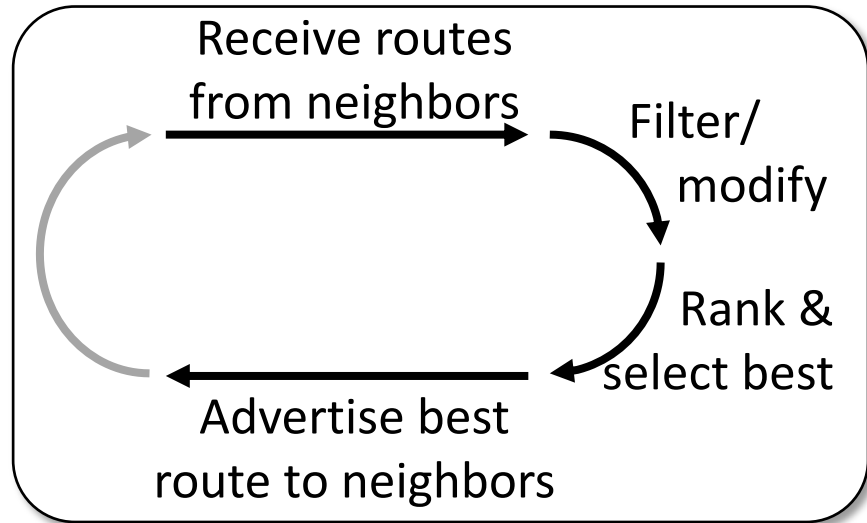
# Modular, three-stage framework





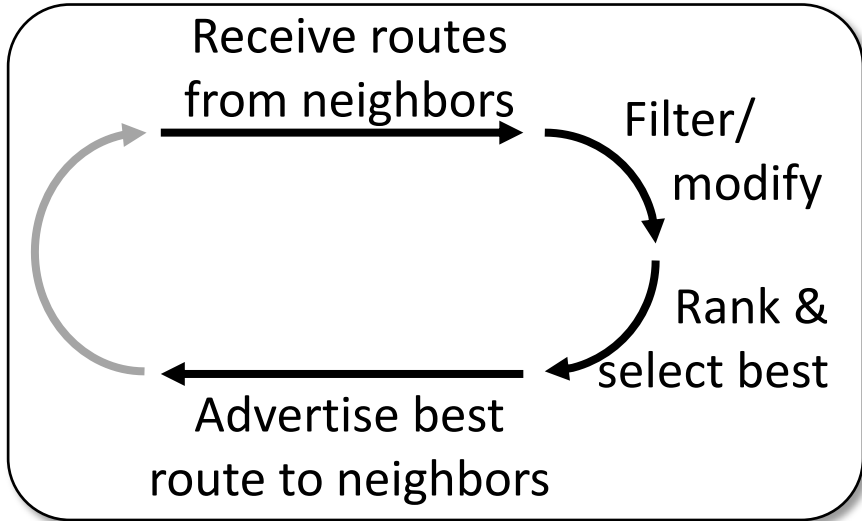
# ① Differential control plane simulation

## Control plane operation

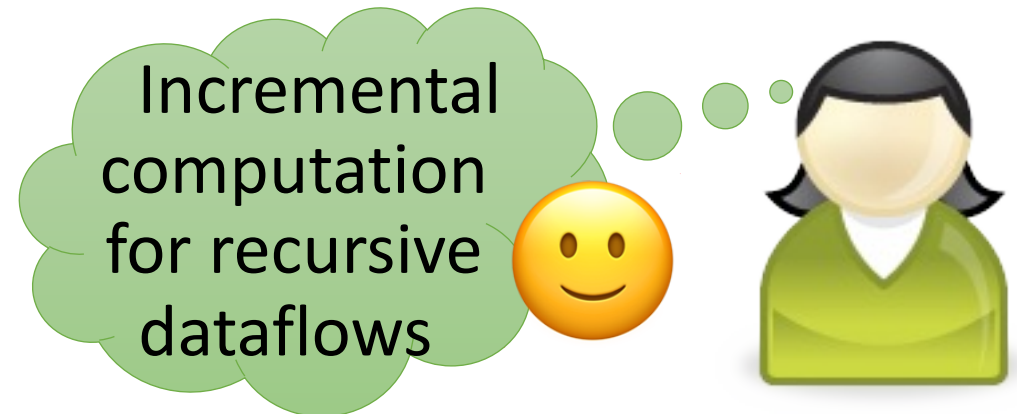
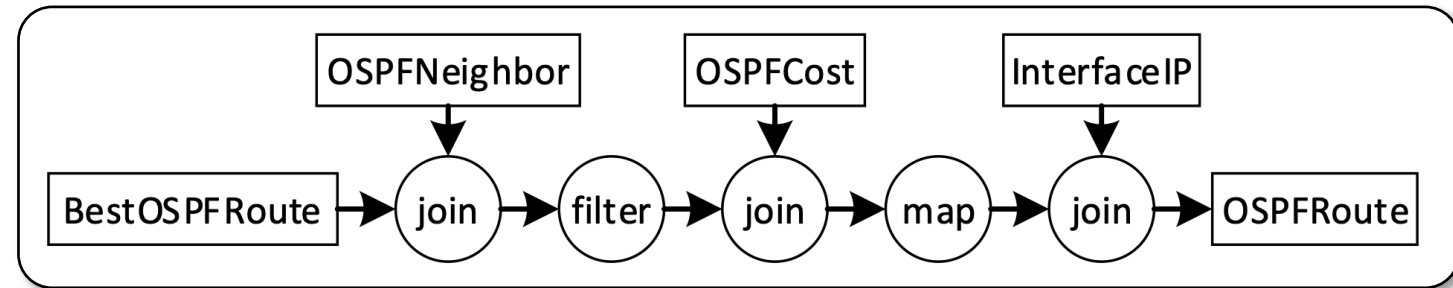


# ① Differential control plane simulation

## Control plane operation

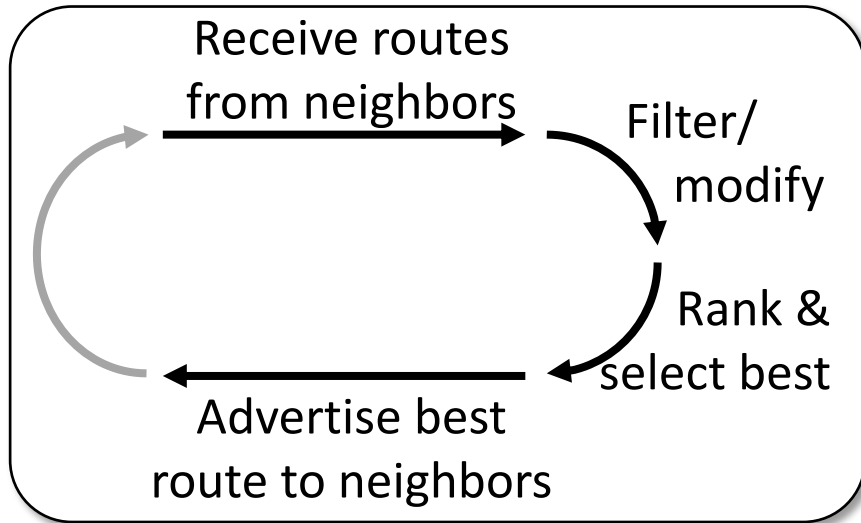


## Modeling with *Differential Dataflow (DD)*

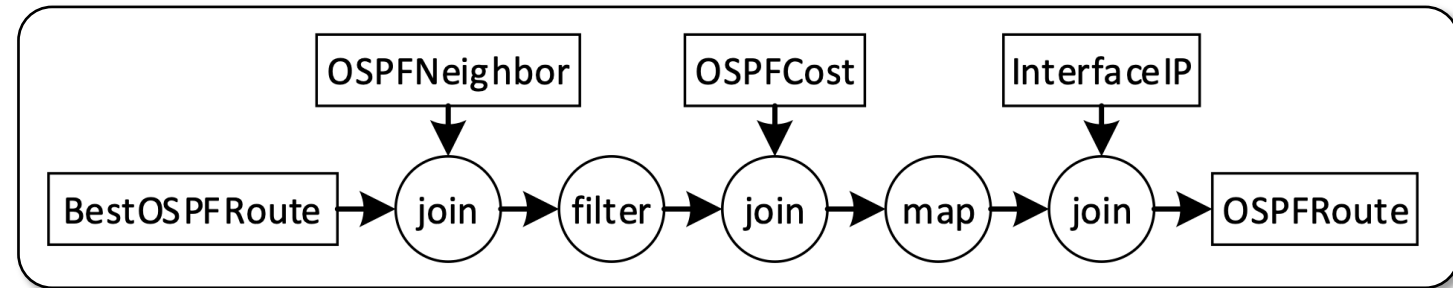


# ① Differential control plane simulation

## Control plane operation



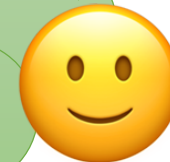
## Modeling with *Differential Dataflow (DD)*



Join, map,  
etc. are too  
low-level

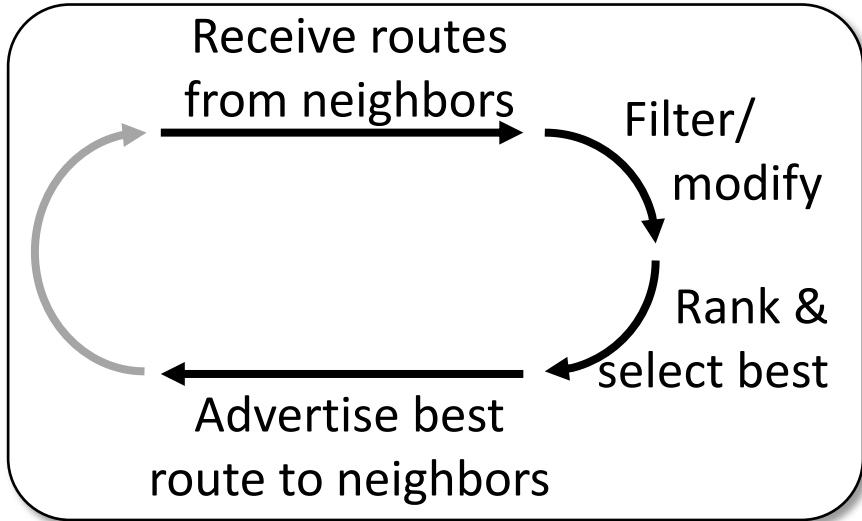


Incremental  
computation  
for recursive  
dataflows

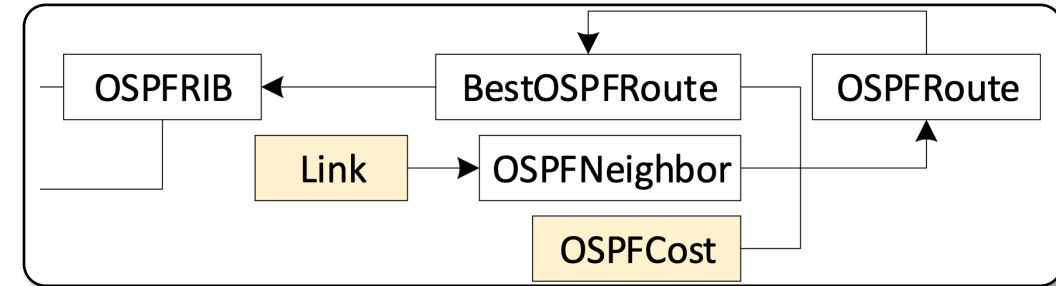


# ① Differential control plane simulation

## Control plane operation



## Modeling with *Differential Datalog (DDlog)*



DDlog model is easy to extend

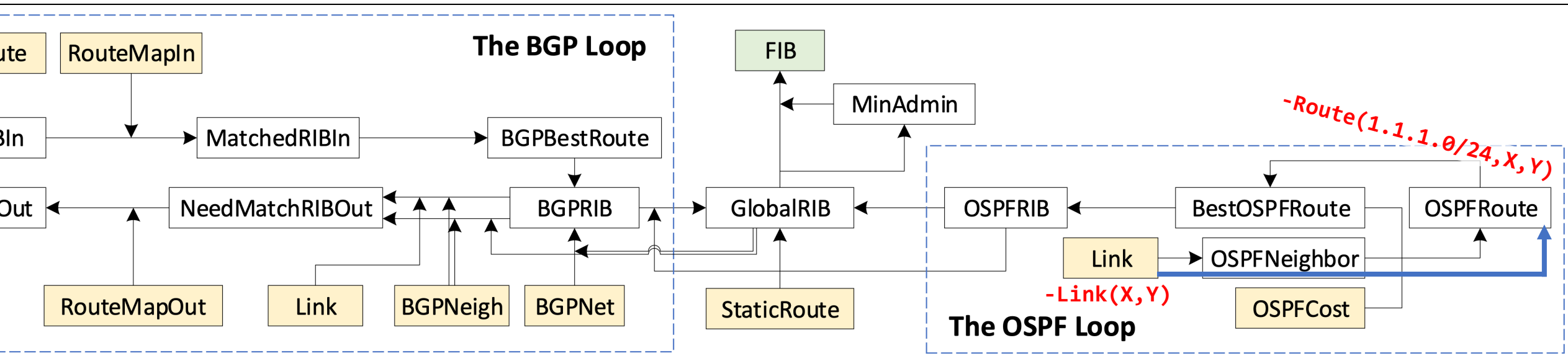


Incremental computation for recursive dataflows

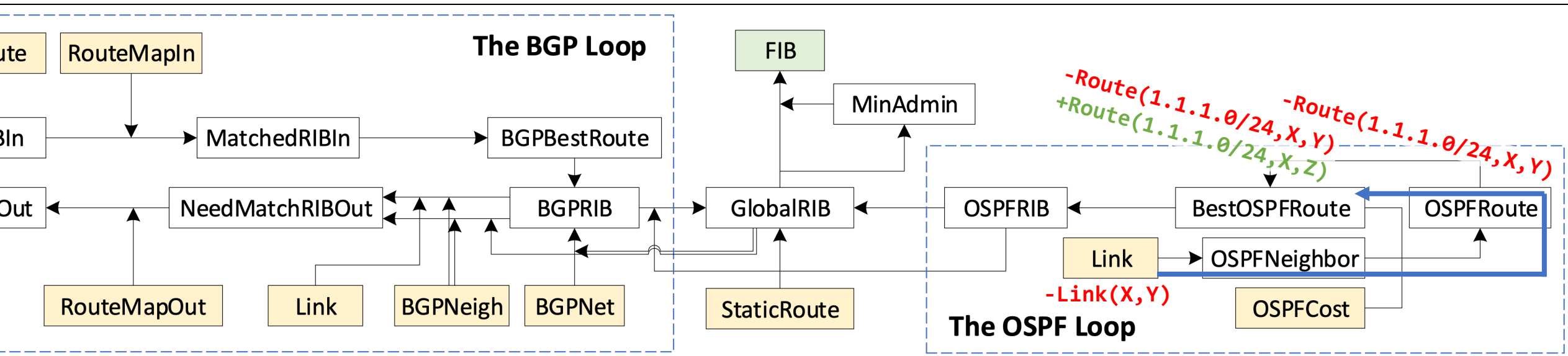




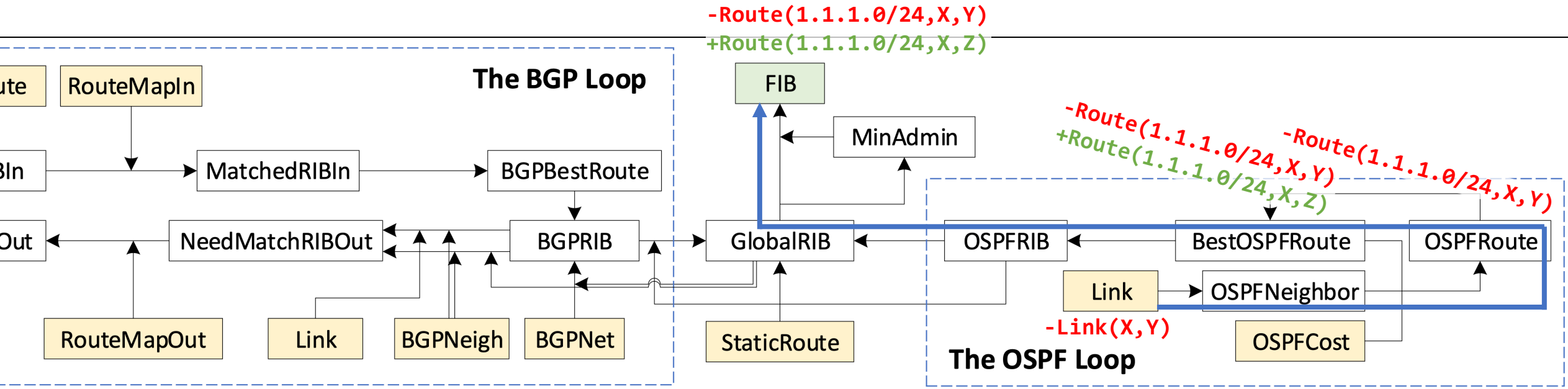
# ① Differential control plane simulation



# ① Differential control plane simulation



# ① Differential control plane simulation

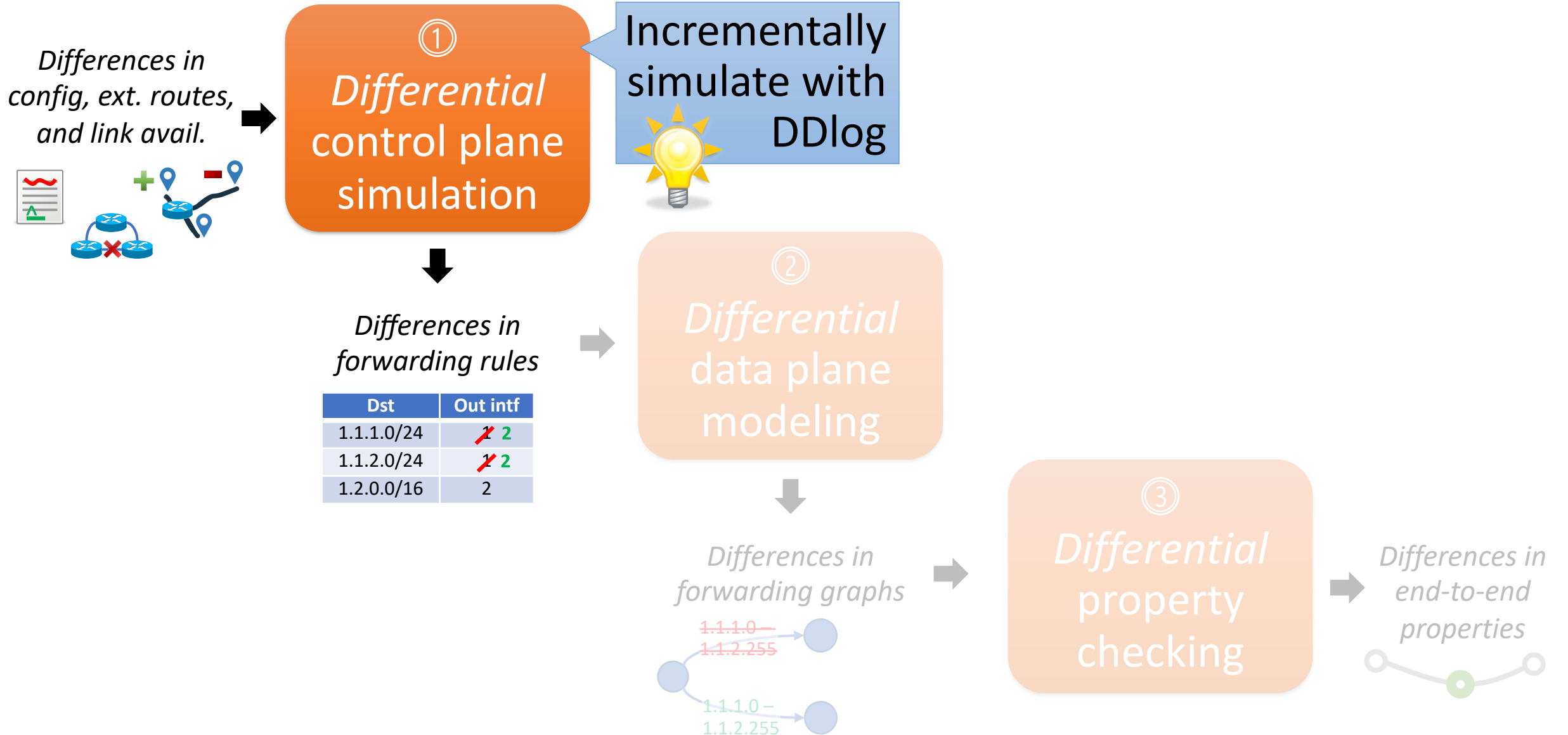




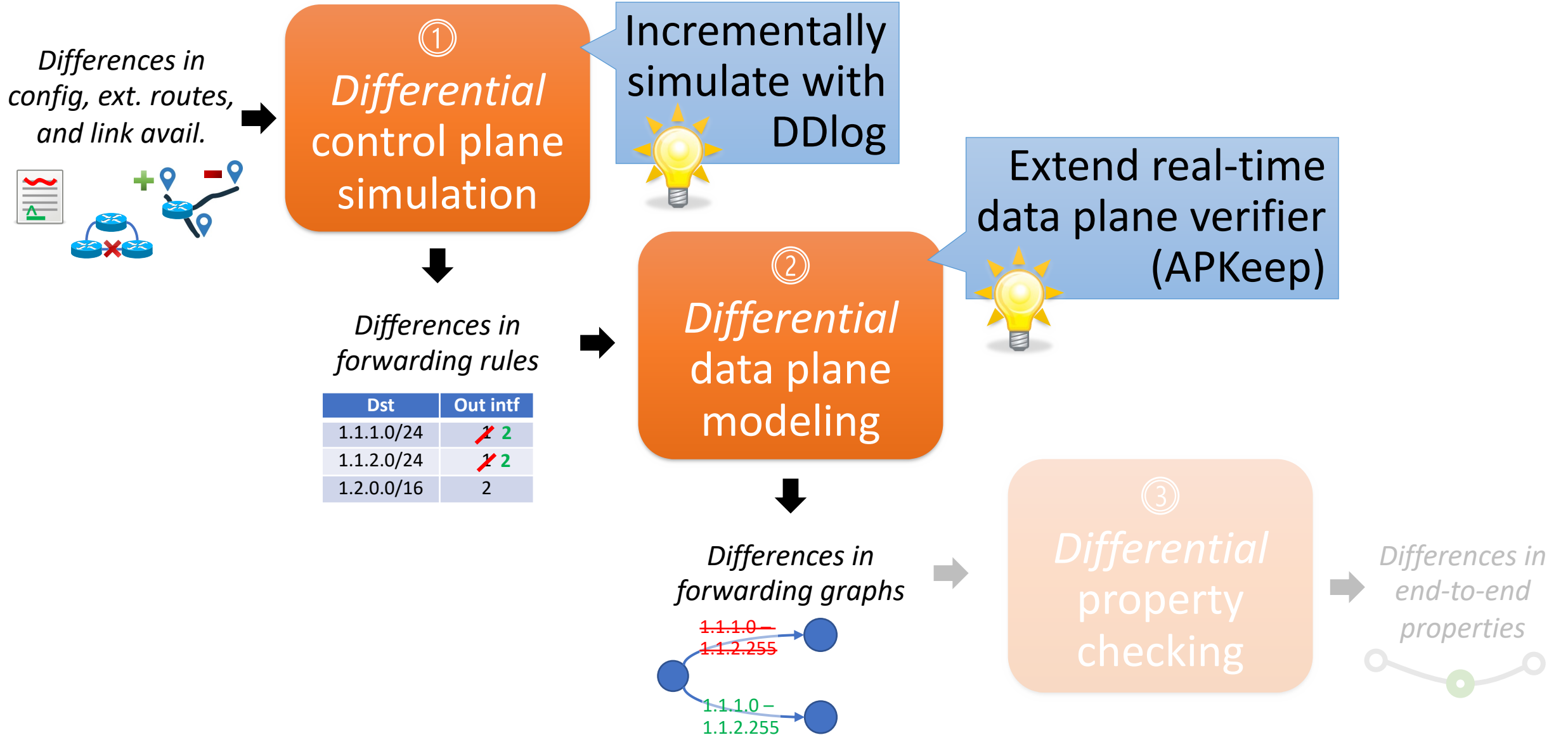




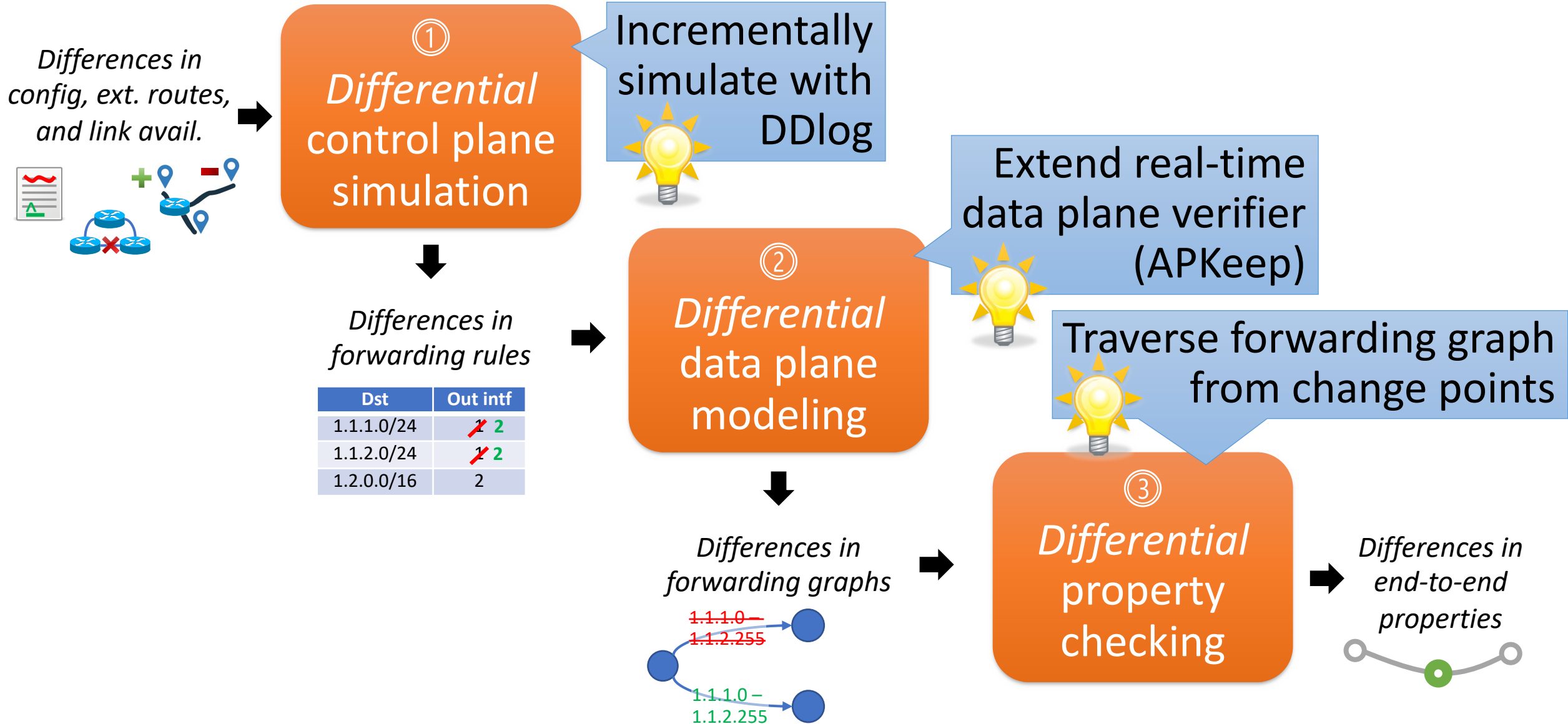
# Modular, three-stage framework



# Modular, three-stage framework

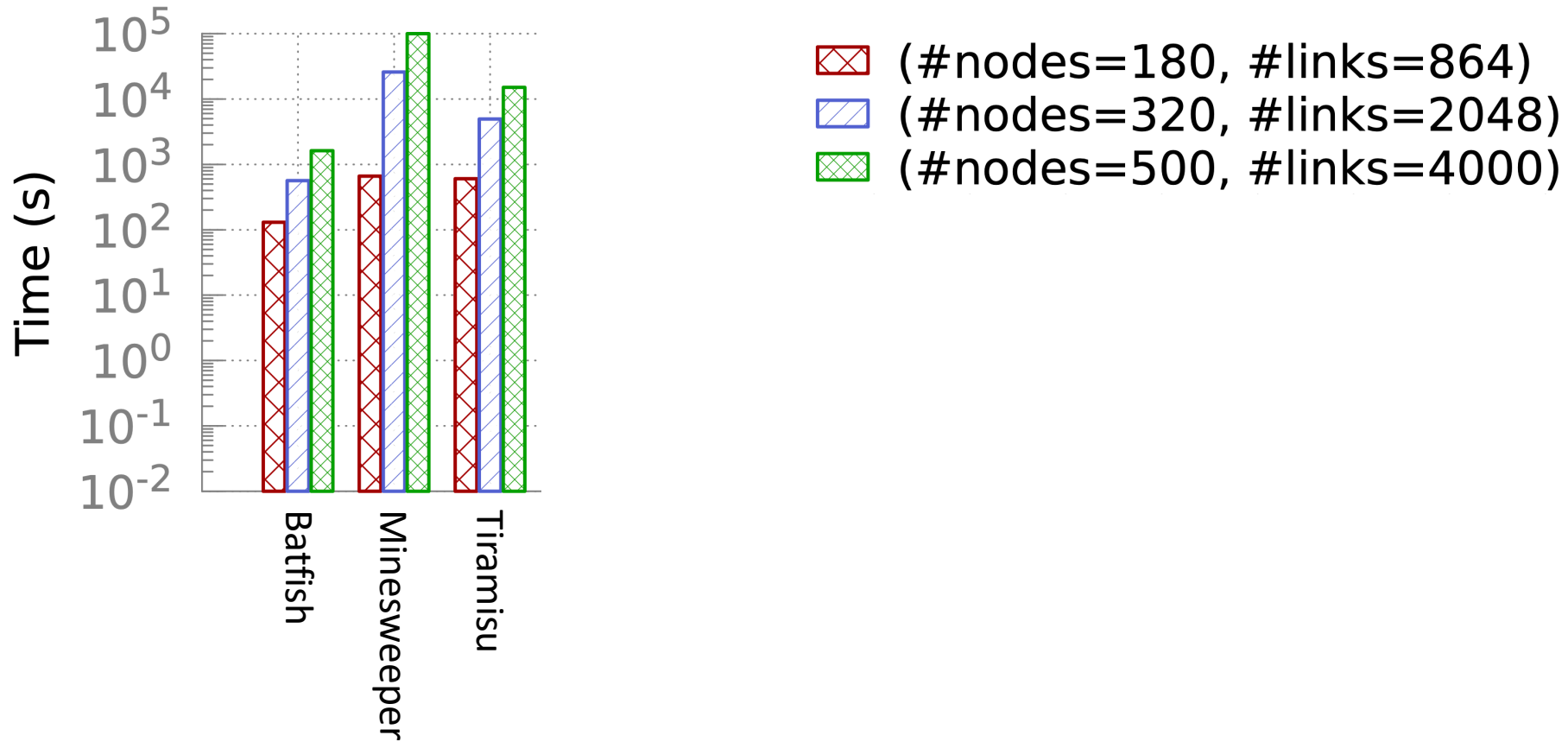


# Modular, three-stage framework



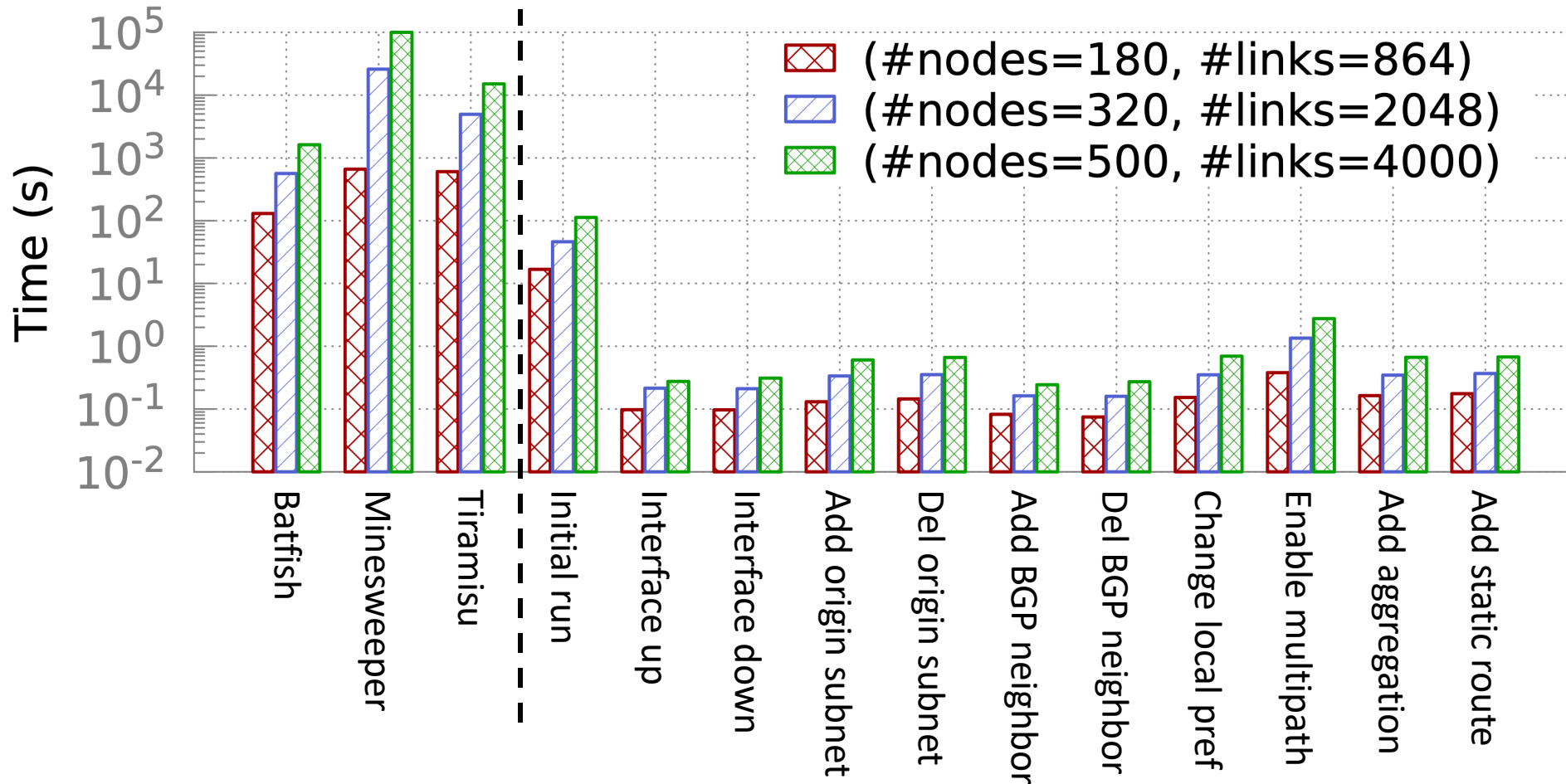
# Evaluation: end-to-end performance

*Synthetic fat tree configurations (BGP) and changes*



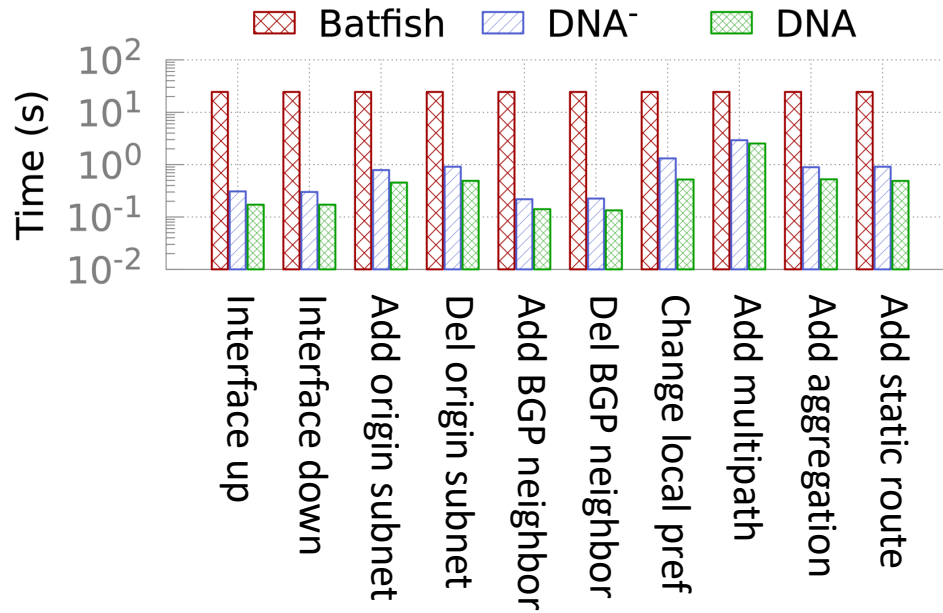
# Evaluation: end-to-end performance

*Synthetic fat tree configurations (BGP) and changes*



# Evaluation: micro-benchmarks

① *Differential control plane simulation*



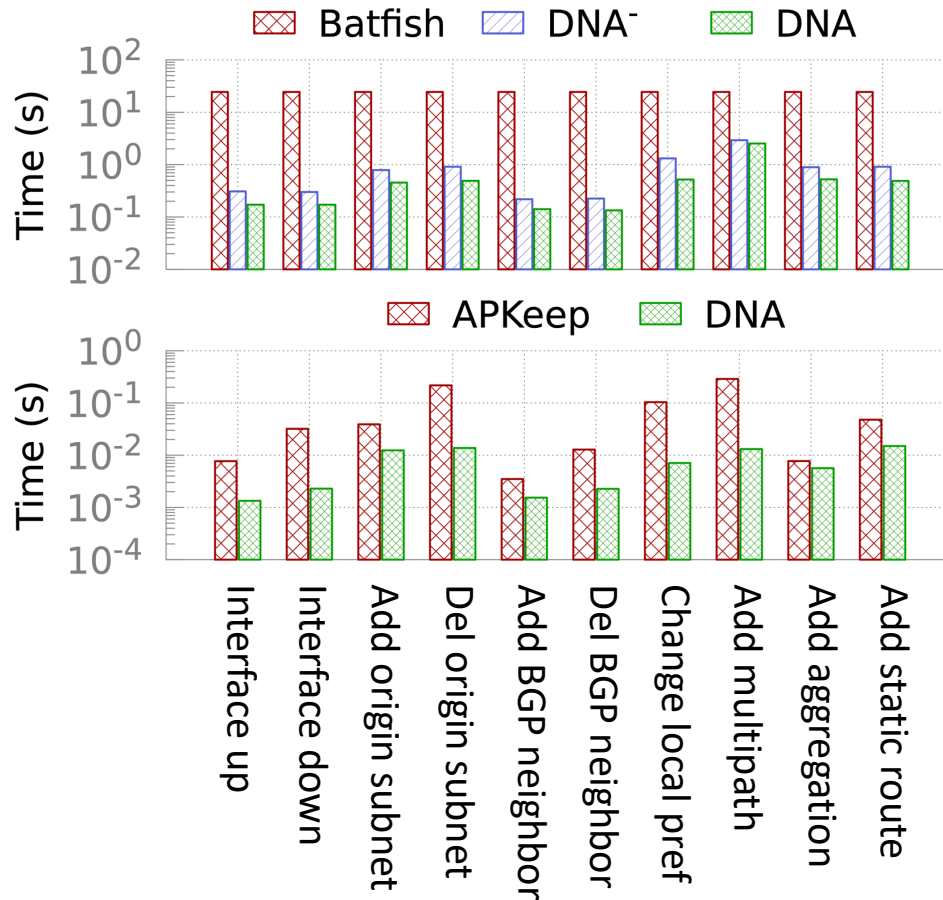


# Evaluation: micro-benchmarks

① *Differential control plane simulation*



② *Differential data plane modeling*



# Evaluation: micro-benchmarks

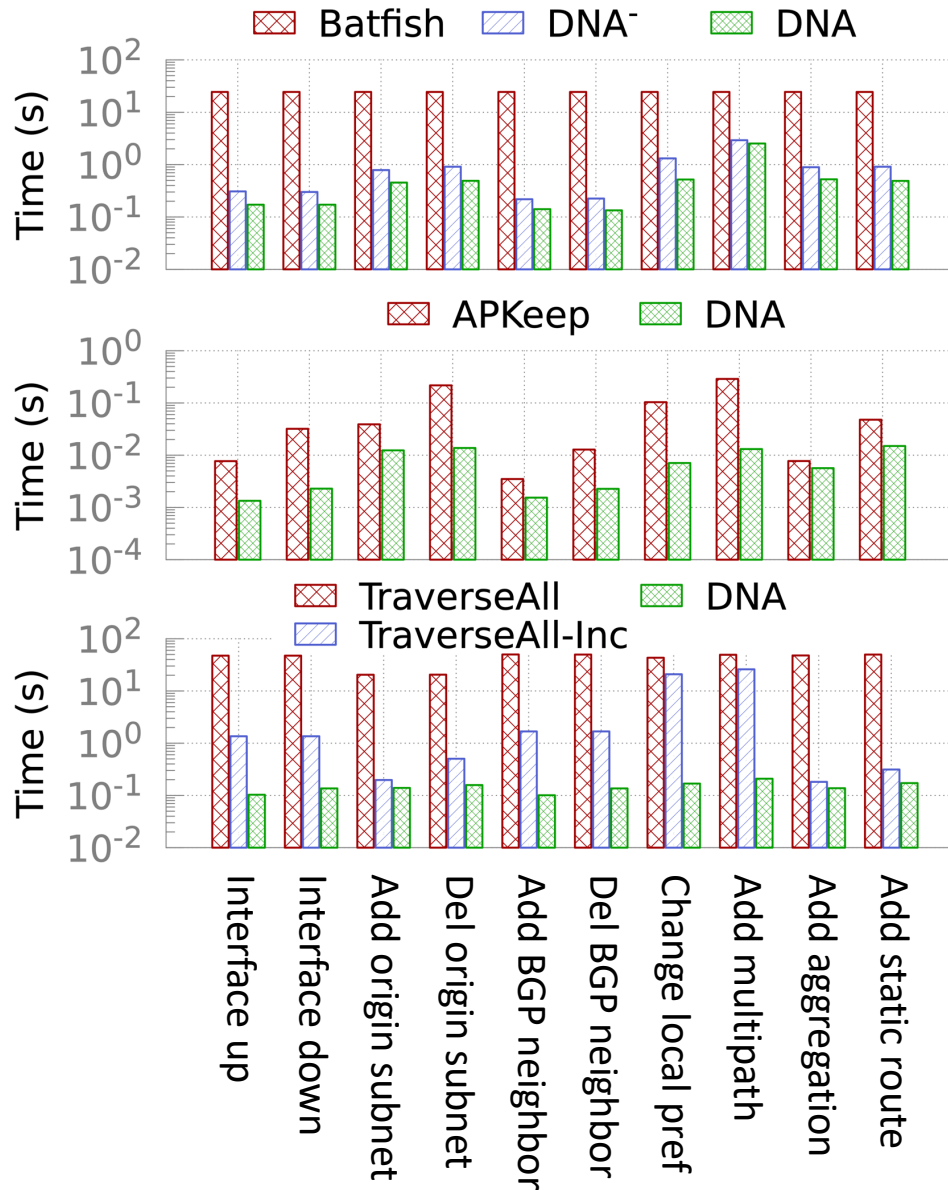
① Differential control plane simulation



② Differential data plane modeling



③ Differential property checking



# Summary

- Networks are frequently in flux; operators need to know whether changes are “safe”
- Invoking existing control plane verifiers is...
  - Inefficient, because changes are often small
  - Difficult, because properties to (re-)verify are not obvious
- Make differences first class citizens
  - Differential control plane simulation using DDlog with custom functions
  - Differential data plane modeling using APKeep with batched updates
  - Differential property checking using optimized forwarding graph traversals