NetMiner : Network Specification Mining with High Fidelity and Scalability

Ning Kang, *Peng Zhang, Hao Li, Sisi Wen,*
Chaoyang Ji, Yongqiang Yang

西安交通大學
XI'AN JIAOTONG UNIVERSITY

HUAWEI

NetMiner

Network Outages are Common

1

Google Accidentally Broke Japan's Internet

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

The New York Times

United Airlines Grounds Flights, Citing Computer Problems

By Christopher Drew

July 8, 2015

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

Xbox Live outage caused by network configuration problem

BY TODD BISHOP on April 15, 2013 at 9:27 am

Home / Cisco Security / Security Advisories and Alerts

Security Activity Bulletin

Misconfigured Router Causes Increased BGP Traffic and Isolated Outages for Internet Services

Google Compute Engine Incident #16015

Networking issue with Google Compute Engine services

Incident began at **2016-08-05 00:54** and ended at **2016-08-05 02:40** (all times are **US/Pacific**).

- **The proportion of outages costing over $100,000 has soared in recent years.** Over 60% of failures result in at least $100,000 in total losses, up substantially from 39% in 2019. The share of outages that cost upwards of $1 million increased from 11% to 15% over that same period.

Massive route leak causes Internet slowdown

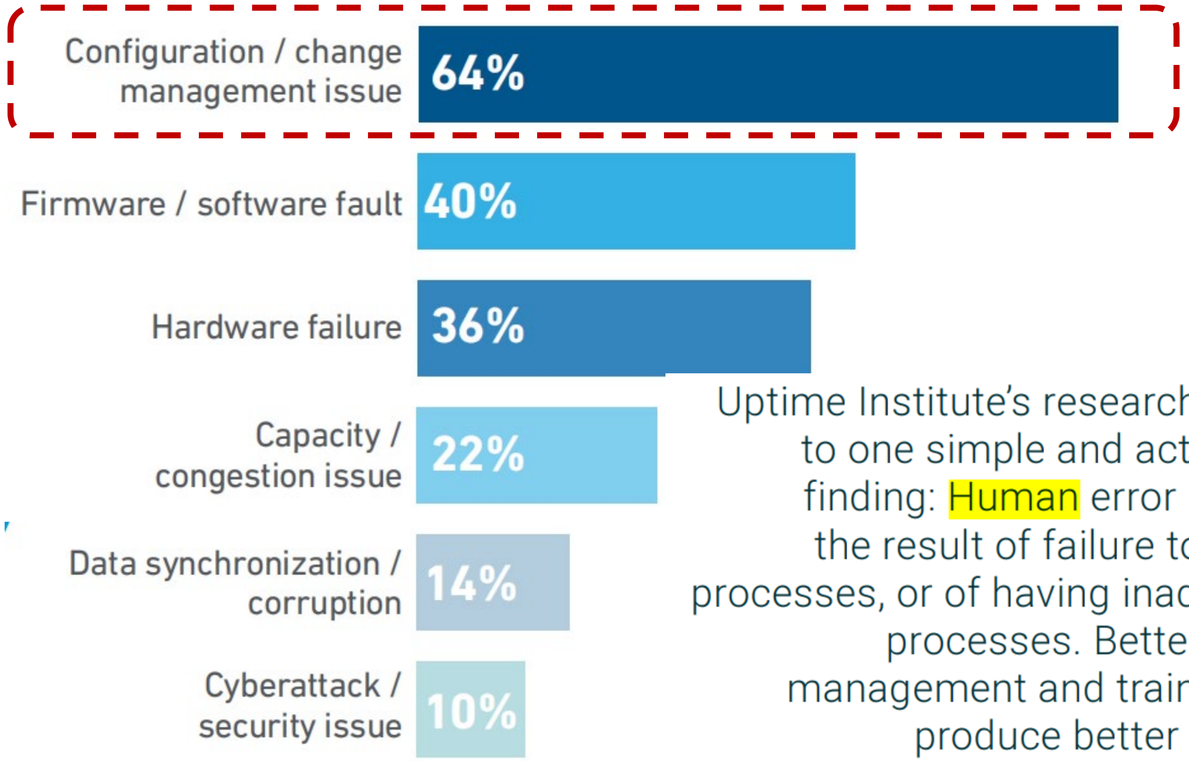
Posted by Andree Toonk – June 12, 2015 – BGP instability – No Comments

Causes of Network Outages

Network outages are mostly caused by human **network management** errors

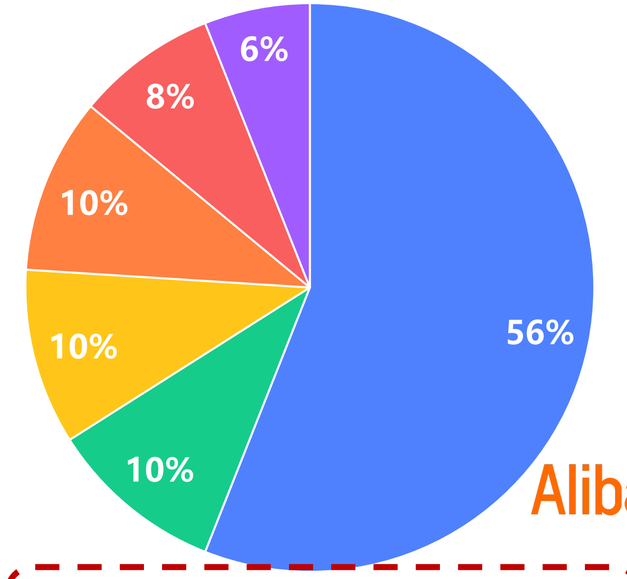
uptime INTELLIGENCE

Annual outages analysis 2023



Uptime Institute's research points to one simple and actionable finding: Human error is often the result of failure to follow processes, or of having inadequate processes. Better focus, management and training will produce better results.

Categories of network outages in 2016 and 2017

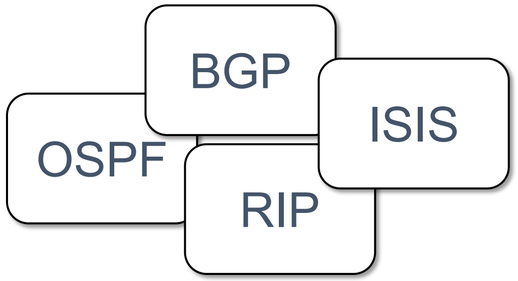


Alibaba Cloud

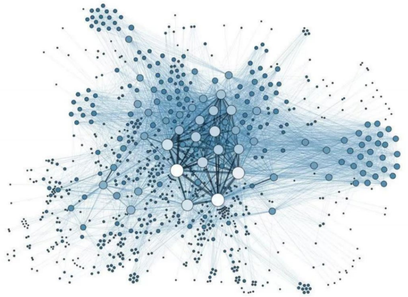
- Configuration Update Error
- Configuration Generation Error
- Human Error
- Device Firmware Bug
- Transient Error
- Unknown

Task1 Understanding Configurations

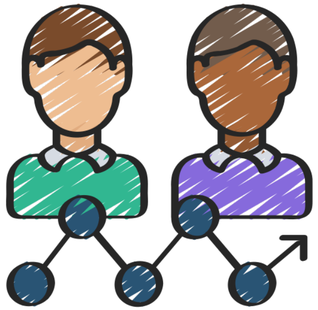
It’s complex for operators to understand a legacy network



Complex routing protocols



Large-scale network topology

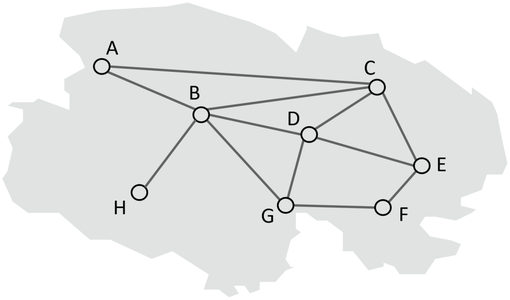


Operators

understand

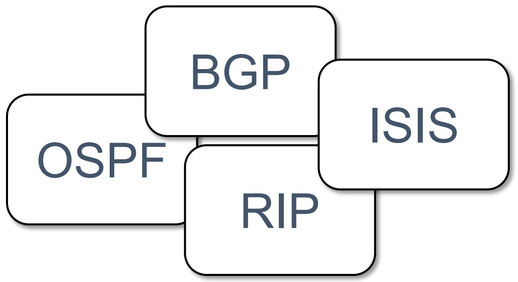
Configurations

download



Physical Networks

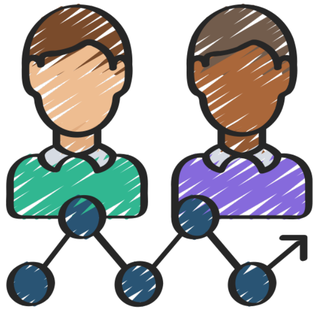
Task2 Updating Configurations



Complex routing protocols

Large-scale network topology

It’s complex to verify unintented side-effects

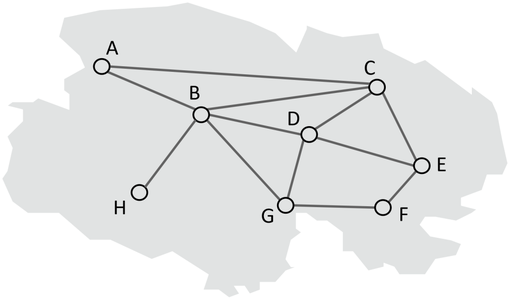


Operators

updates

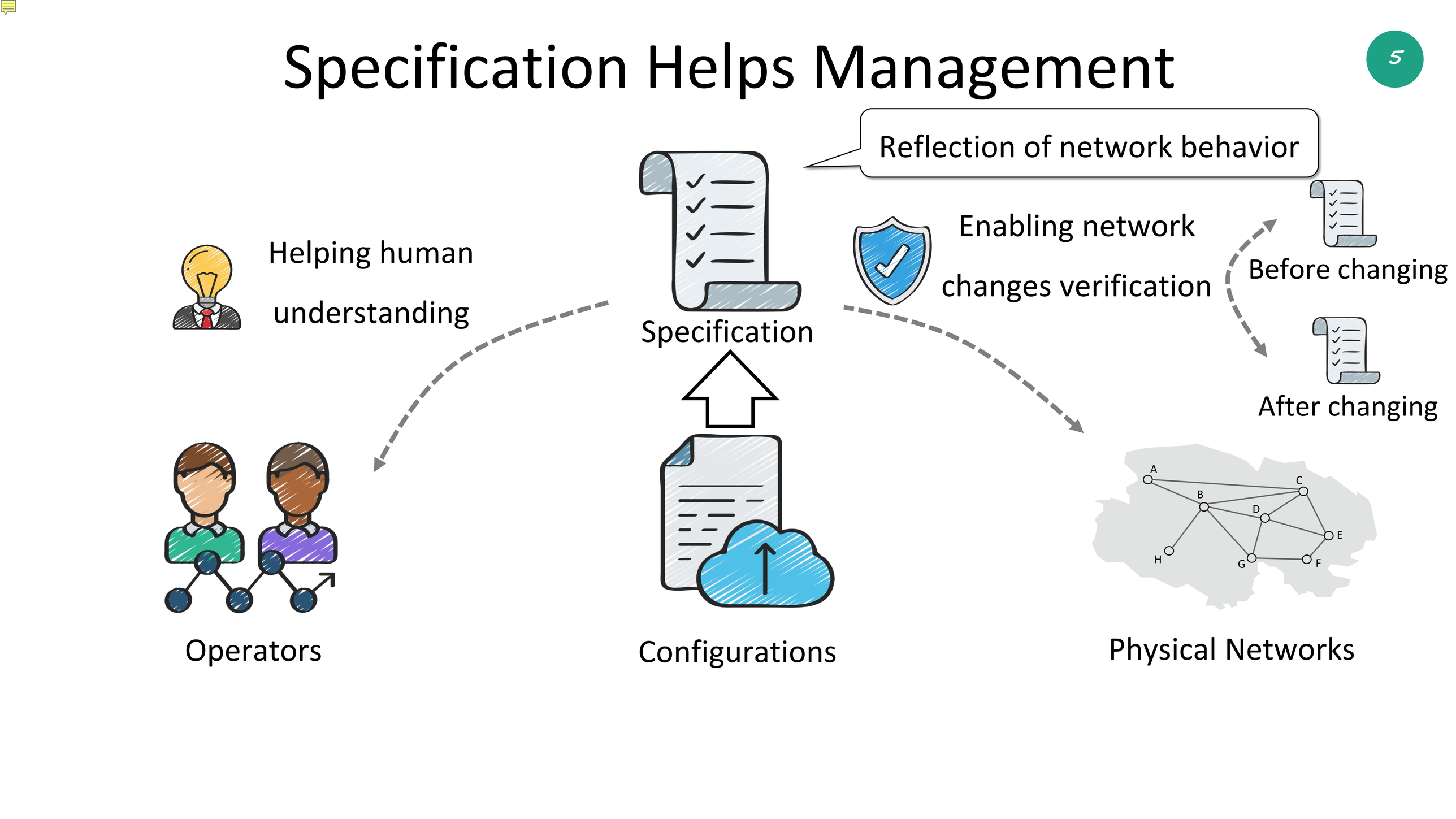
Configurations

upload



Physical Networks

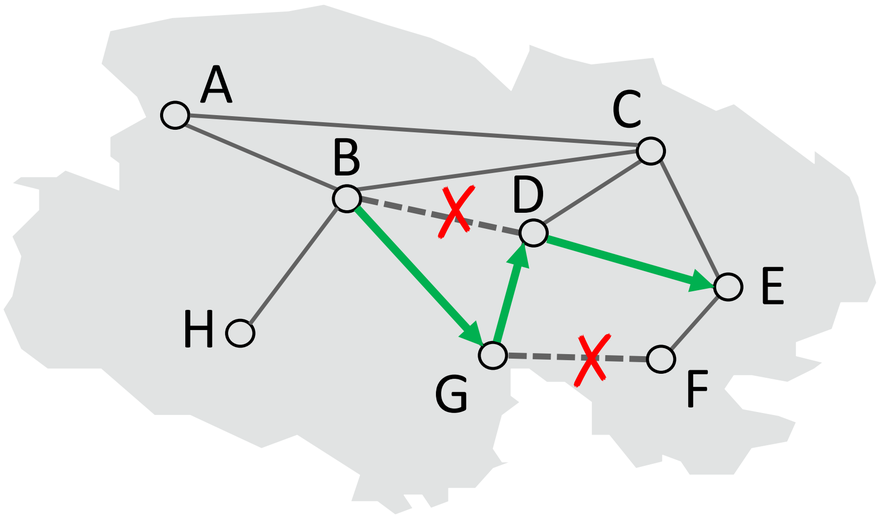
Specification Helps Management



Formal Specification

Specification : the set of all properties and their failure tolerance

All properties	Failure tolerance
reachability(**A**, **B**)	t = 1
waypoint(**A**, **D**, **E**)	t = 1
▸ reachability(**B**, **E**)	t = 2
loadbalance(**B**, **F**)	t = 2



B to **E** is still reachable
after failing any **two** links

Current Approach

Configurations

Config2Spec

Data Plane Analysis

Control Plane Verification

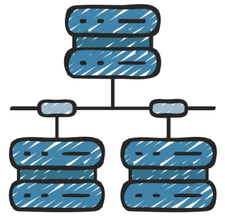
Specification

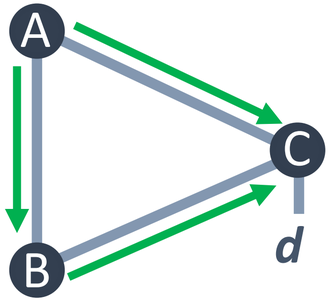
Difficulty to Obtain Specification

Fidelity Specification should reflect the same network behavior governed by configurations

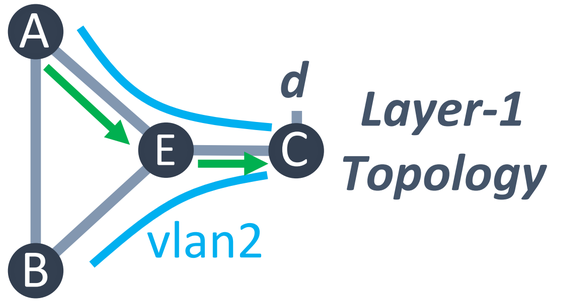
1 Realistic failure model

Physical link failures instead of logic link failures





reachability(A , *d*) : t = **1** ✗



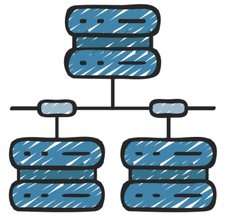
reachability(A , *d*) : t = **0** ✓

Difficulty to Obtain Specification

Fidelity Specification should reflect the same network behavior with configurations

1 Realisitic failure model

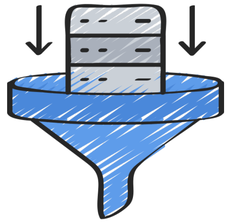
Physical link failures instead of logic link failures



2 Vendor feature support

VRFs, router filters matching multiple tags,

protocol- or port-based packet filters, ...



Difficulty to Obtain Specification

Scalability Obtaining network specification should support large-scale networks

Real DCN : **130** devices

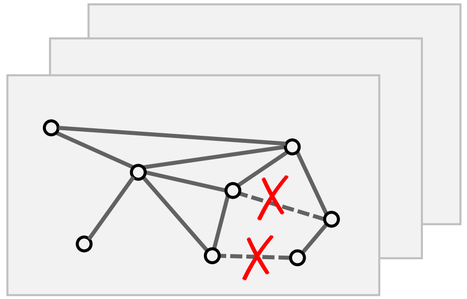
5314 physical links

1 A large number of properties

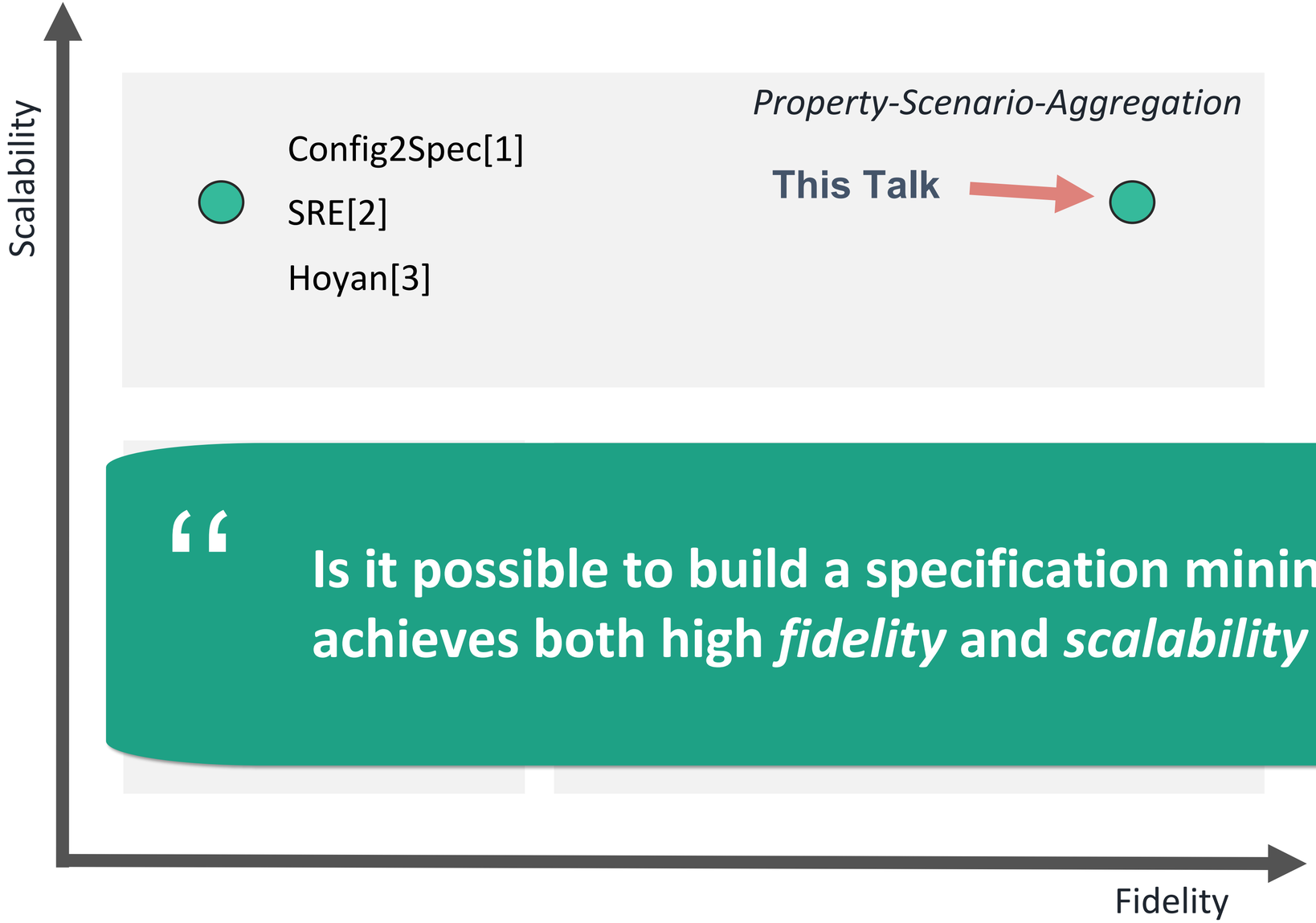
reachability is **127754**, loadbalance is **116627**, ...

2 A large number of failure scenarios

$C_{5314}^{1} \approx 10^{4}$ $C_{5314}^{2} \approx 10^{7}$ $C_{5314}^{3} \approx 10^{11}$ ...



Tools Available for Mining Specification



[1] Birkner et al, NSDI'20

[2] Zhang et al, SIGCOMM'22

[3] Ye et al, SIGCOMM'20

[4] Steffen et al, SIGCOMM'20

[5] Abhashkumar et al, NSDI'20

[6] [illegible] SIGCOMM'16

[7] [illegible]

[8] [illegible]20

[9] [illegible]19

[10] [illegible]5

[11] Liu et al, SOSP'17

"Is it possible to build a specification mining tool that achieves both high *fidelity* and *scalability* ?"

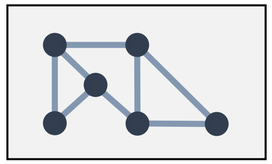

NetMiner

Insights

- *Use pure simulation-based approach to ensure high fidelity*

Insights

- *Use pure simulation-based approach to ensure high fidelity*



Layer-1 Topologies
(failure scenarios)

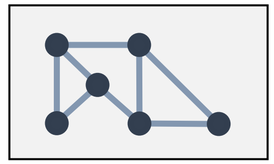
Select relevant scenarios

NetMiner

Specification Mining

Insights

- *Use pure simulation-based approach to ensure high fidelity*



Layer-1 Topologies
(failure scenarios)

Control Plane Simulation

NetMiner

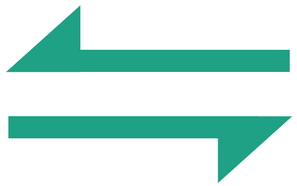
Specification Mining

Insights

- *Use pure simulation-based approach to ensure high fidelity*

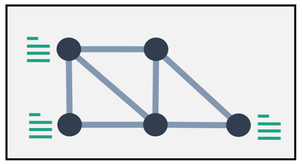
Layer-1 Topologies
(failure scenarios)

Control Plane Simulation



NetMiner

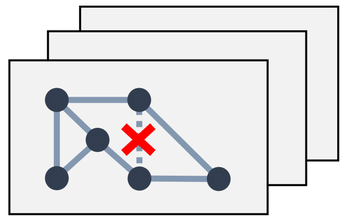
Specification Mining



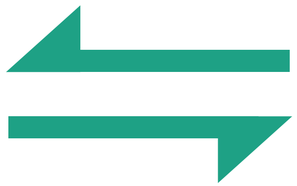
Routing Tables

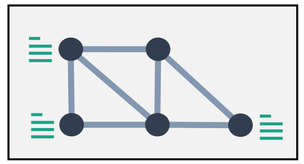
Insights

- *Use pure simulation-based approach to ensure high fidelity*



Layer-1 Topologies
(failure scenarios)





Routing Tables

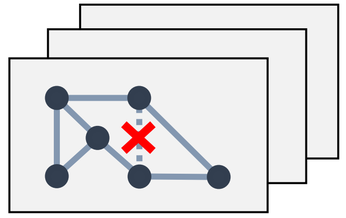
Select new scenarios based on Ribs

Insights

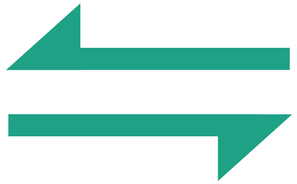
- *Use pure simulation-based approach to ensure high fidelity*

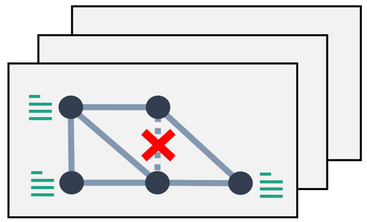
Generate Ribs for each scenarios

Control Plane Simulation



Layer-1 Topologies
(failure scenarios)





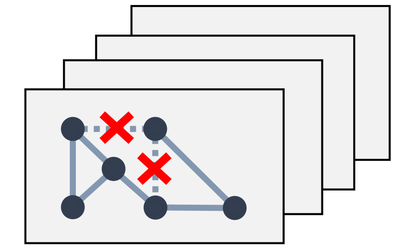
Routing Tables

NetMiner

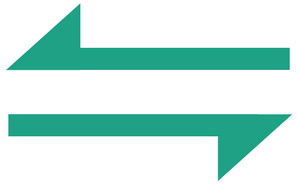
Specification Mining

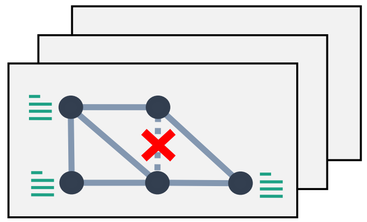
Insights

- *Use pure simulation-based approach to ensure high fidelity*

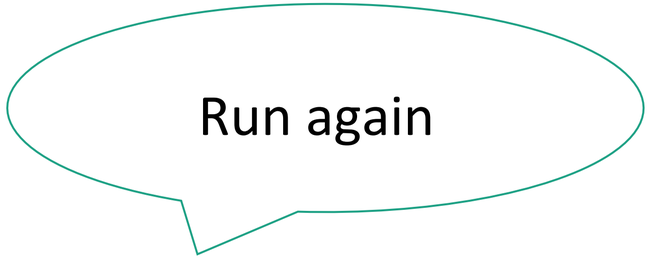


Layer-1 Topologies
(failure scenarios)





Routing Tables

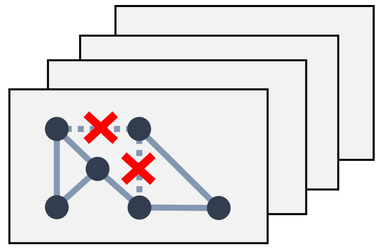


Insights

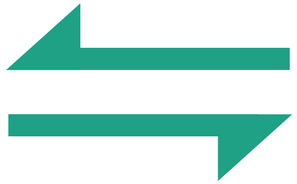
- *Use pure simulation-based approach to ensure high fidelity*

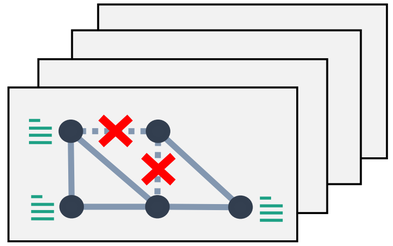
Run again

Control Plane Simulation



Layer-1 Topologies
(failure scenarios)





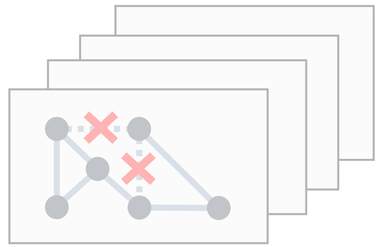
Routing Tables

NetMiner

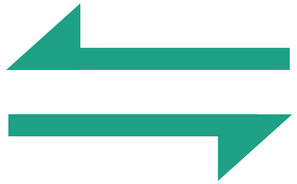
Specification Mining

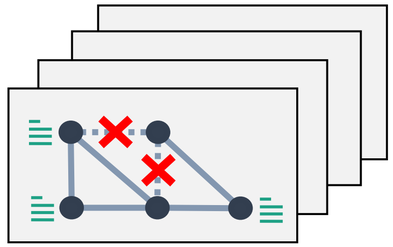
Insights

- *Use pure simulation-based approach to ensure high fidelity*



Layer-1 Topologies
(failure scenarios)





Routing Tables

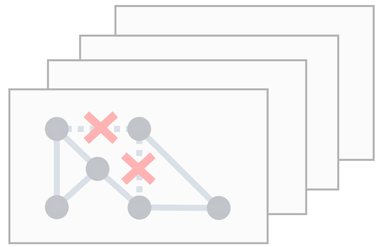
Terminate in a certain round

Insights

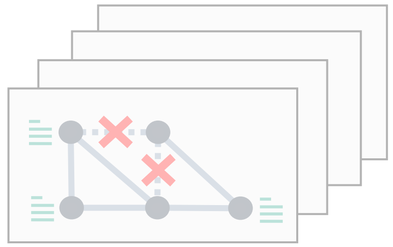
- *Use pure simulation-based approach to ensure high fidelity*

Data Plane Generation

Control Plane Simulation



Layer-1 Topologies
(failure scenarios)



Routing Tables

Failure Scenarios Selection

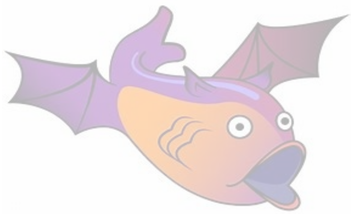
NetMiner

Specification Mining

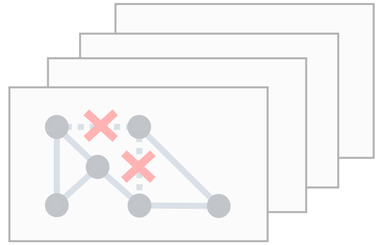
Challenge 1

- How to efficiently generate failure scenarios ?

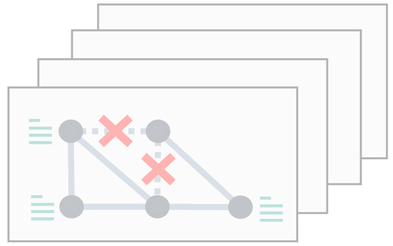
Data Plane Generation



Control Plane Simulation



Layer-1 Topologies
(failure scenarios)



Routing Tables

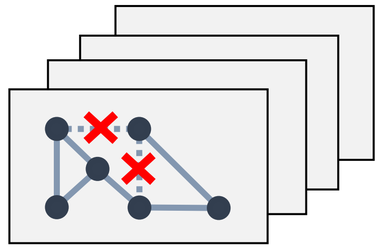
Failure Scenarios Selection

NetMiner

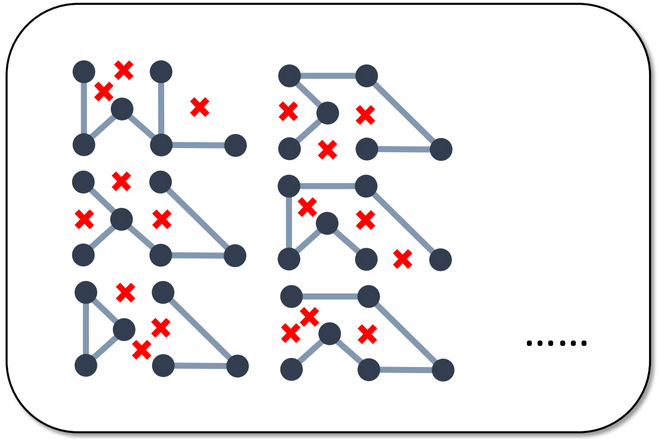
Specification Mining

Challenge 1

- How to efficiently generate failure scenarios ?



Layer-1 Topologies
(failure scenarios)



~ 10K properties

Failure Scenarios Identification

Specification Mining

2 A large number of failure scenarios

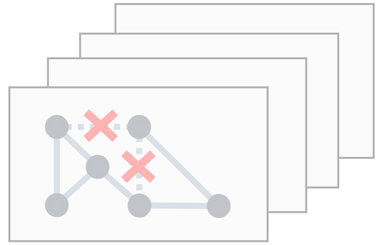
1 A large number of properties

Challenge 2

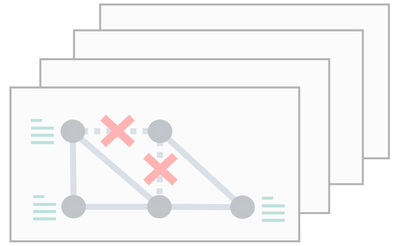
- How to make layer-3 topology generation scalable ?

Data Plane Generation

Control Plane Simulation



Layer-1 Topologies
(failure scenarios)



Routing Tables

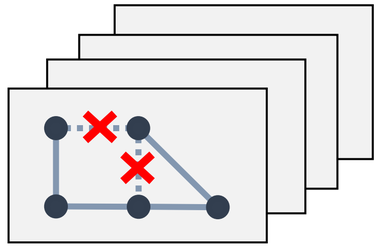
Failure Scenarios Selection

NetMiner

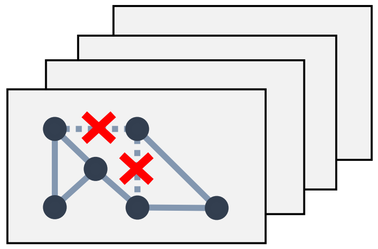
Specification Mining

Challenge 2

- How to make layer-3 topology generation scalable ?



Layer-3 Topologies
(failure scenarios)



Layer-1 Topologies
(failure scenarios)

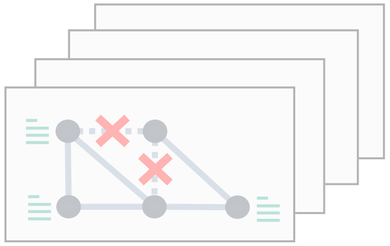
Failure Scenarios Selection

Layer3 Topology Generation

Control Plane Simulation

NetMiner

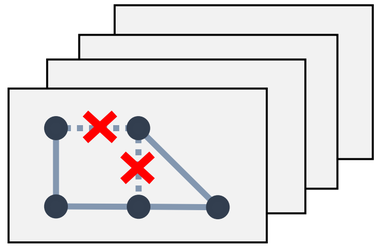
Specification Mining



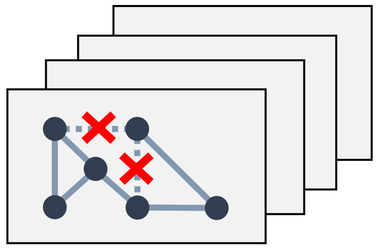
Routing Tables

Challenge 2

- How to make layer-3 topology generation scalable ?



Layer-3 Topologies
(failure scenarios)

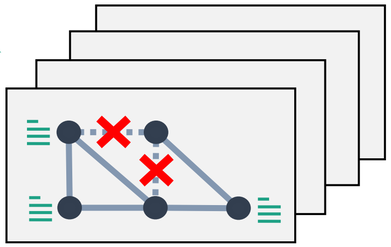


Layer-1 Topologies
(failure scenarios)

Failure Scenarios Selection

Layer3 Topology Generation | Routing Table Generation

Control Plane Simulation



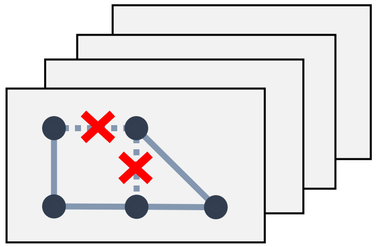
Routing Tables

NetMiner

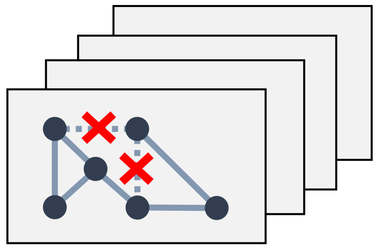
Specification Mining

Challenge 2

- How to make layer-3 topology generation scalable ?



Layer-3 Topologies
(failure scenarios)



Layer-1 Topologies
(failure scenarios)

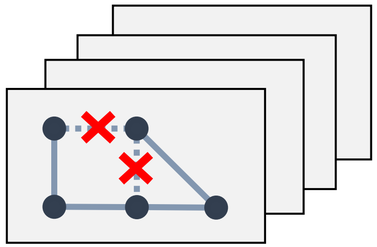
Layer3 Topology Generation | Routing Table Generation

Control Plane Simulation

- *A large number of **devices**, physical **ports** as well as **vlans***
- *run Batfish on two real data center networks*

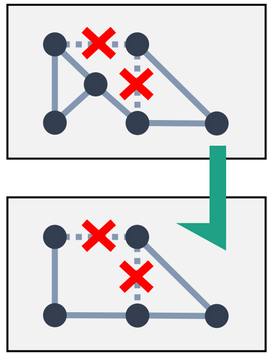
	Layer3 topology computation time	Routing tables generation time
DC1	7.8s	3.1s
DC2	192.7s	0.54s

Challenges



- How to efficiently generate failure scenarios ?

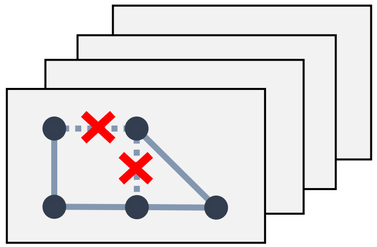
  the total number of failure scenarios



- How to make layer-3 topology generation scalable ?

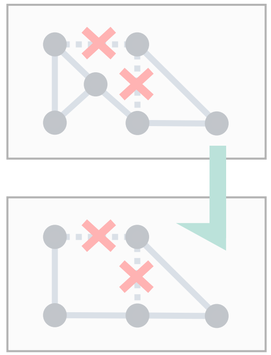
  the time of analyzing a single failure scenarios

Challenges



- How to efficiently generate failure scenarios ?

the total number of failure scenarios



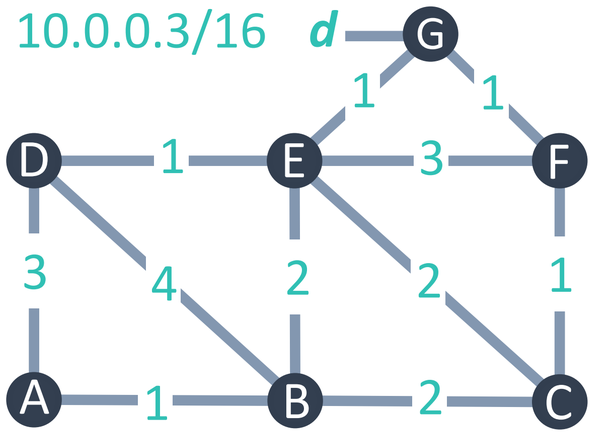
- How to make layer-3 topology generation scalable ?

the time of analyzing a single failure scenario

General Scenarios Aggregation (GSA)

General Scenarios Aggregation

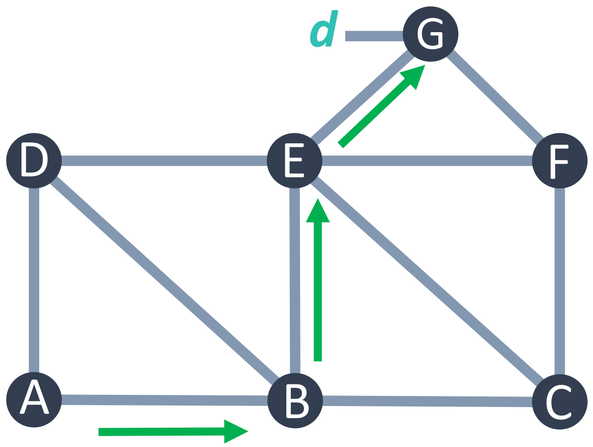
- **Example : OSPF Network**

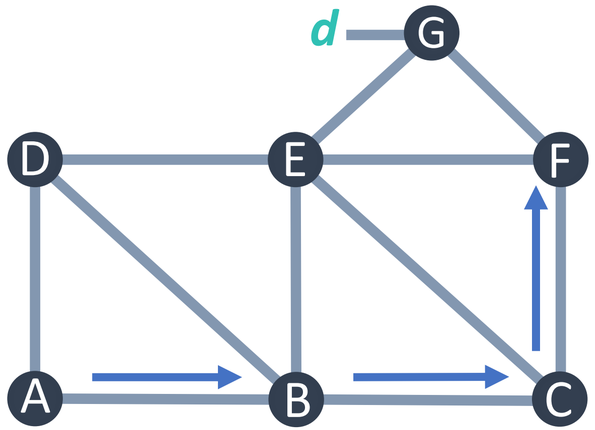


Reachability(A, *d*) t=?

Reachability(A, F) t=?

General Scenarios Aggregation

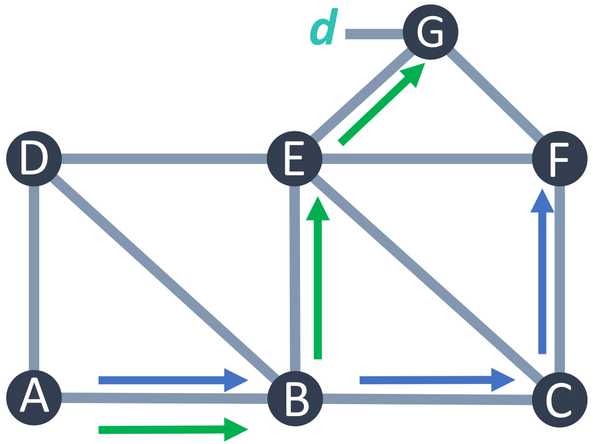




Reachability(A, ***d***) t=?

Reachability(A, F) t=?

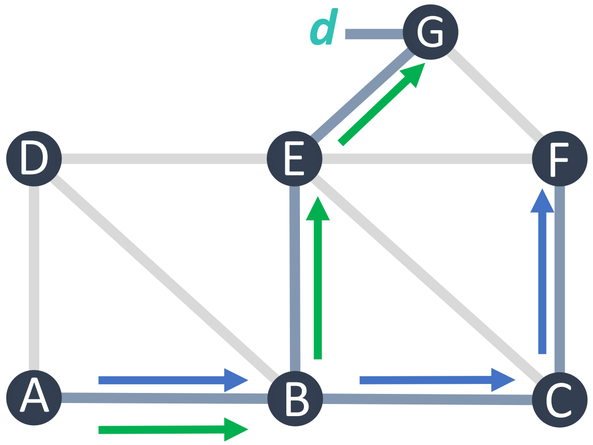
General Scenarios Aggregation



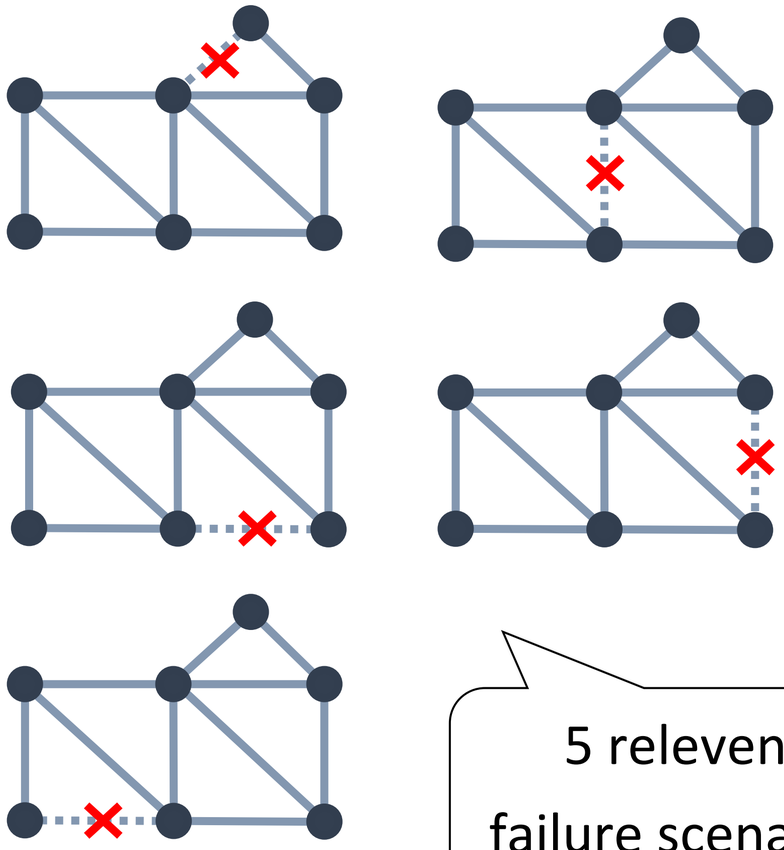
Reachability(A, ***d***) t=?

Reachability(A, F) t=?

General Scenarios Aggregation



hot links



5 relevent failure scenarios

Reachability(A, ***d***) t=?

Reachability(A, F) t=?

General Scenarios Aggregation

- ***Identifying hot links*** for single property without fidelity loss

Reachability(Ⓐ, ***d***) t=?

Netdice identify hot links by custom algorithm

- ***Aggregating failure scenarios*** between properties

Reachability(Ⓐ, Ⓕ) t=?

General Scenarios Aggregation

- ***Identifying hot links*** for single property without fidelity loss

Reachability(Ⓐ, ***d***) t=?

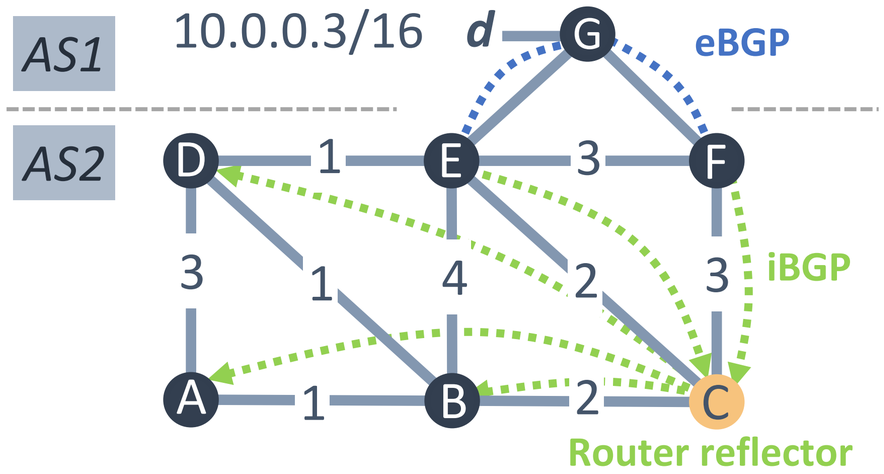
Netdice identify hot links by custom algorithm

- ***Aggregating failure scenarios*** between properties

Reachability(Ⓐ, Ⓕ) t=?

Identify Hot Links

▸**Example : BGP Network**



Identify hot links only based on ***Routing Tables***

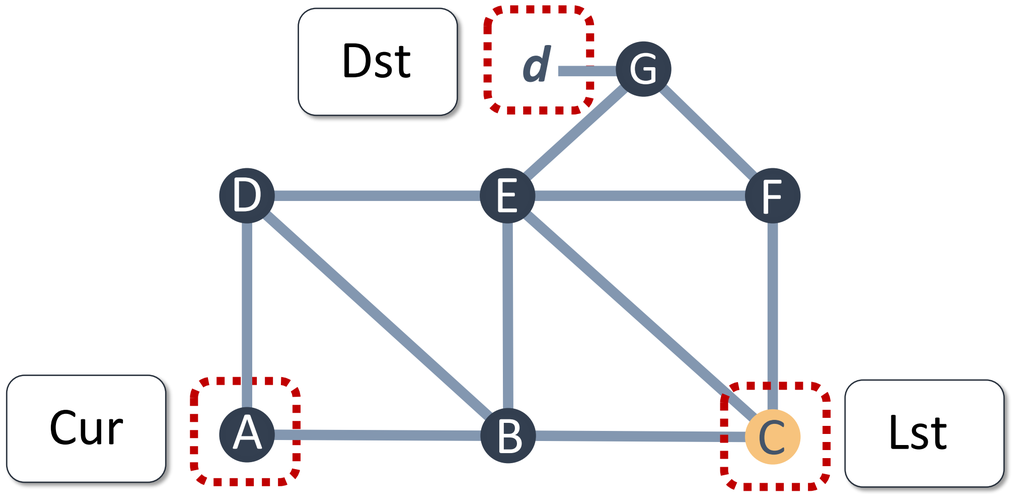
Hot Links = Routing Paths + Forwarding Paths

G → A routing Path

Reachability(A, 10.0.0.3/16, t=?)

Identify Hot Links

▸**Example : BGP Network**



G → A routing Path

Reachability(A, 10.0.0.3/16, t=?)

Main RIB of A

Dst	Next Hop	Learn From	Type
d	E	C	iBGP
E	B	B	OSPF
…	…	…	…

iBGP RIB of A

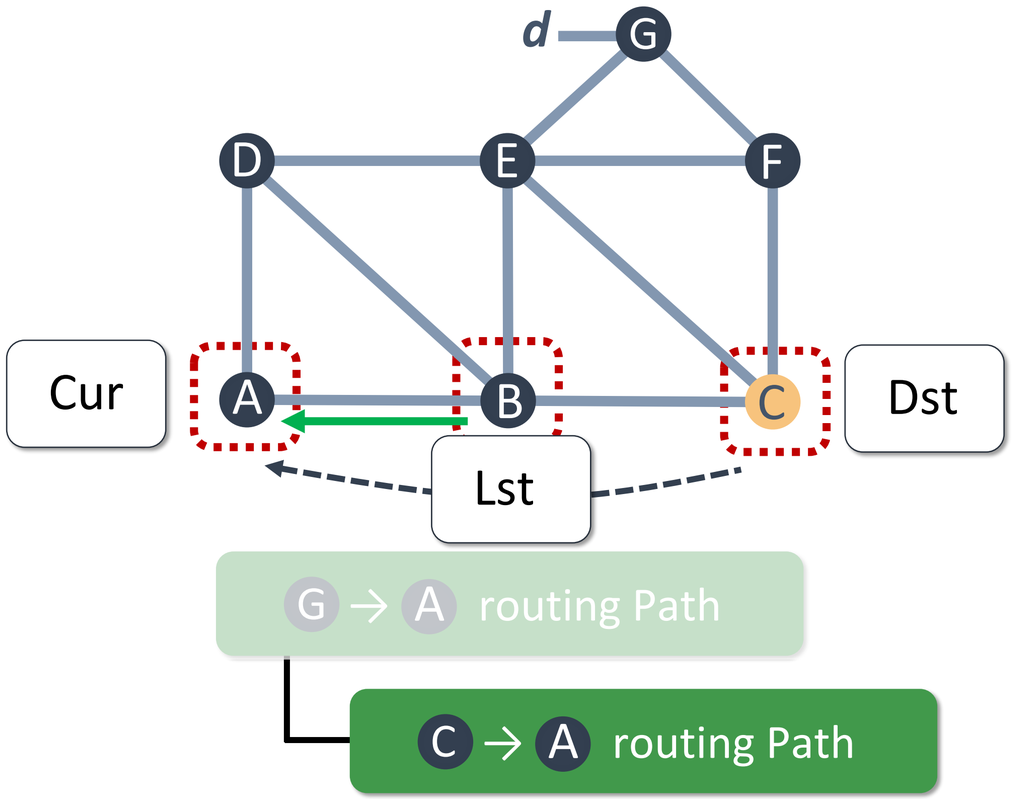
Dst	Next Hop	Learn From	Flag
d	E	C	recursive
G	E	C	recursive
…	…	…	…

OSPF RIB of A

Dst	Next Hop	Learn From	Flag
C	B	B	direct
D	D	D	direct
…	…	…	…

Identify Hot Links

▸**Example : BGP Network**



Main RIB of A

Dst	Next Hop	Learn From	Type
d	E	C	iBGP
E	B	B	OSPF
…	…	…	…

iBGP RIB of A

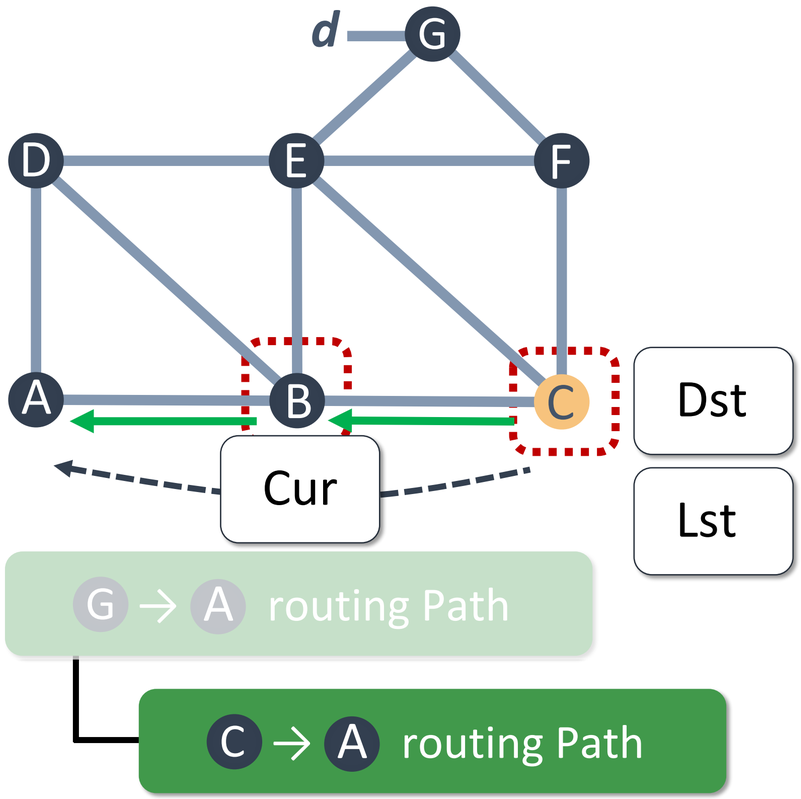
Dst	Next Hop	Learn From	Flag
d	E	C	recursive
G	E	C	recursive
…	…	…	…

OSPF RIB of A

Dst	Next Hop	Learn From	Flag
C	B	B	direct
D	D	D	direct
…	…	…	…

Identify Hot Links

▸**Example : BGP Network**



Main RIB of B

Dst	Next Hop	Learn From	Type
C	C	C	OSPF
d	D	B	Static
…	…	…	…

iBGP RIB of B

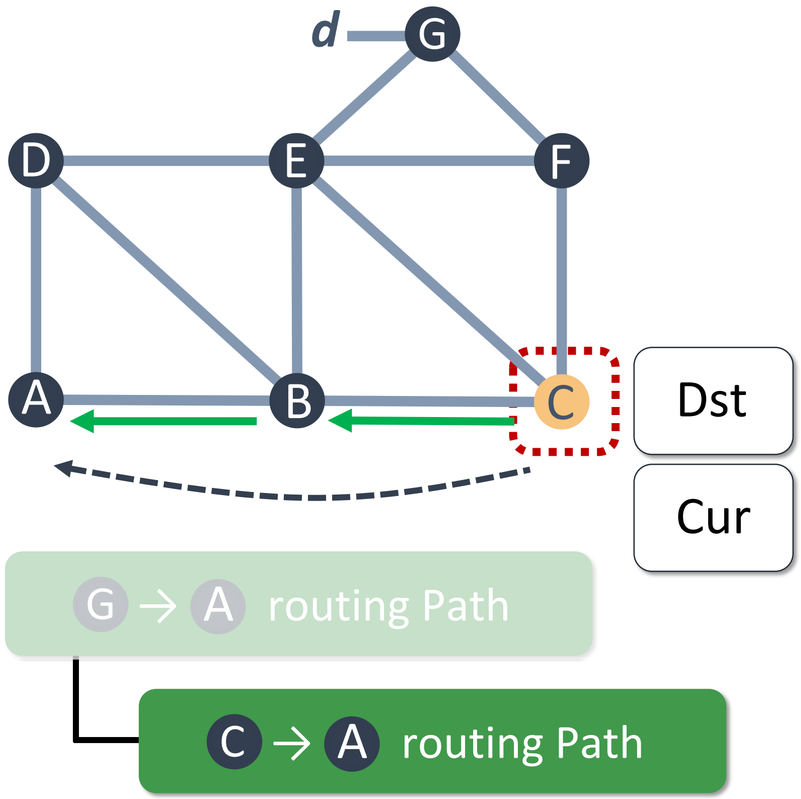
Dst	Next Hop	Learn From	Flag
d	E	C	recursive
G	E	C	recursive
…	…	…	…

OSPF RIB of B

Dst	Next Hop	Learn From	Flag
C	C	C	direct
D	D	D	direct
…	…	…	…

Identify Hot Links

▸**Example : BGP Network**



Main RIB of C

Dst	Next Hop	Learn From	Type
d	E	E	iBGP
F	F	F	OSPF
…	…	…	…

iBGP RIB of C

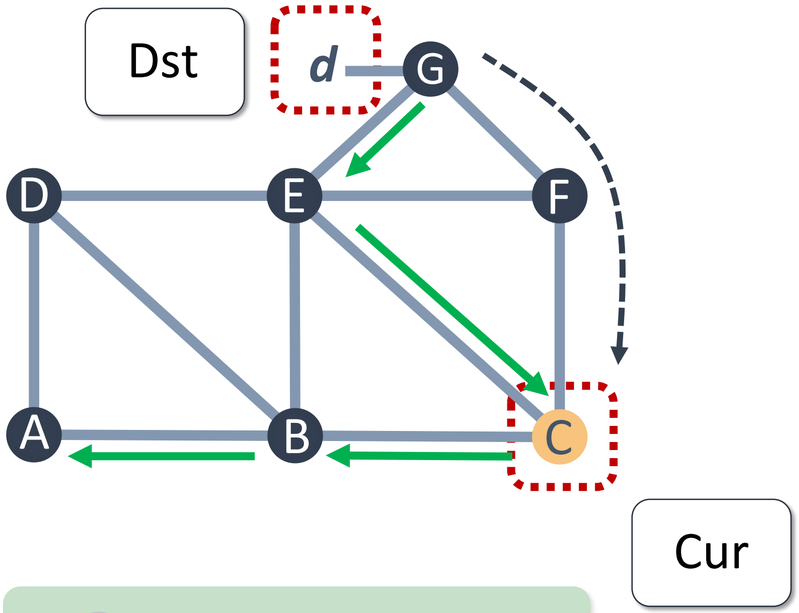
Dst	Next Hop	Learn From	Flag
d	E	E	recursive
G	E	C	recursive
…	…	…	…

OSPF RIB of C

Dst	Next Hop	Learn From	Flag
E	E	E	direct
D	B	B	direct
…	…	…	…

Identify Hot Links

- **Example : BGP Network**



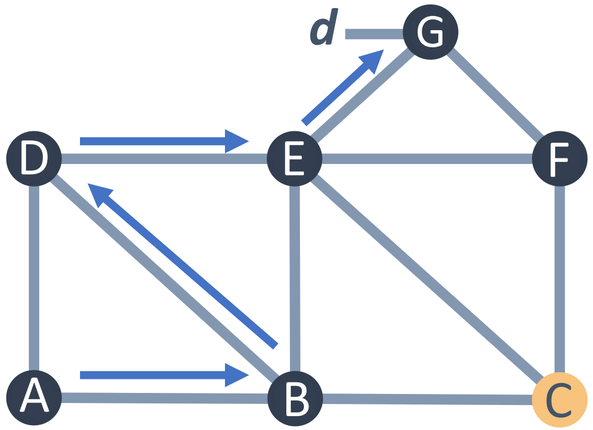
G → A routing Path

C → A routing Path

G → C routing Path

Identify Hot Links

- **Example : BGP Network**



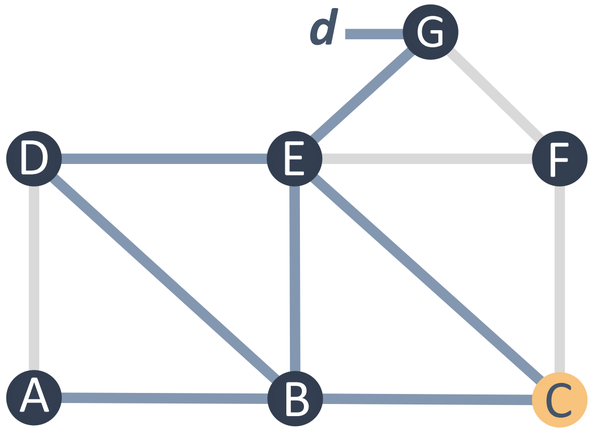
G → A routing Path

A → G forwarding Path

Reachability(A, 10.0.0.3/16, t=?)

Identify Hot Links

- **Example : BGP Network**



Generate hot links only
based on ***Routing tables***

G → A routing Path

A → G forwarding Path

Reachability(A, 10.0.0.3/16, t=?)

General Scenarios Aggregation

- ***Identifying hot links*** for single property without fidelity loss

Reachability(Ⓐ, ***d***) t=?

Netdice identify hot links by custom algorithm

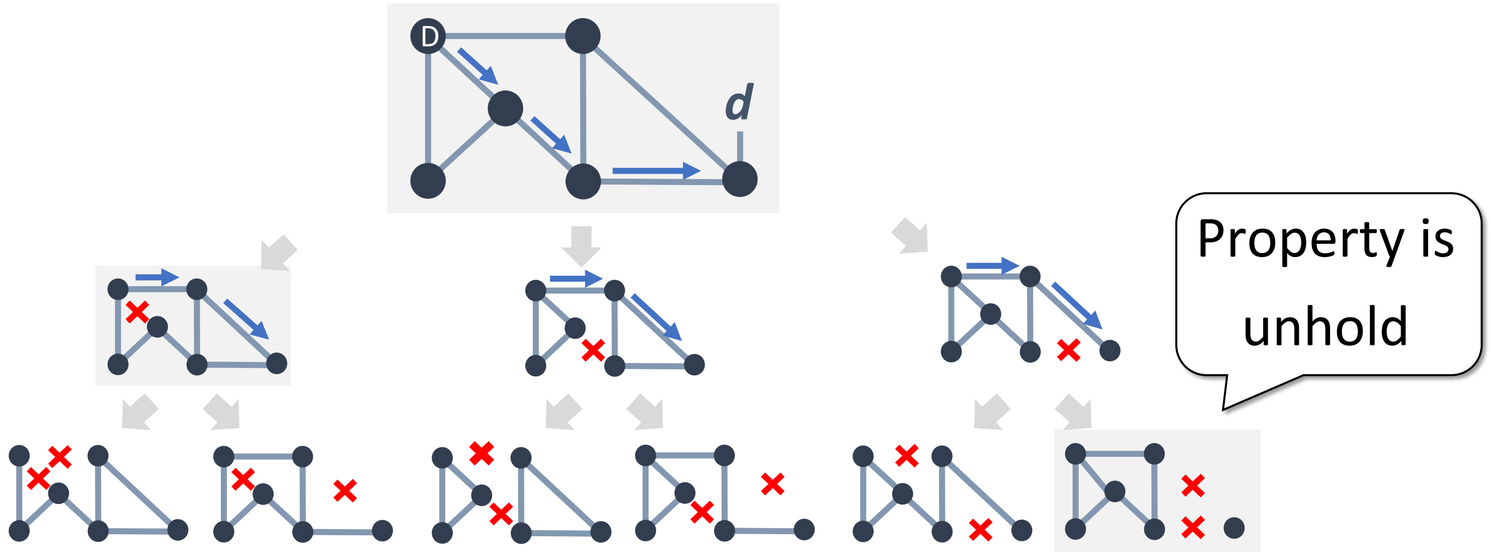
- ***Aggregating failure scenarios*** between properties

Reachability(Ⓐ, Ⓕ) t=?

Failure Scenarios for Single Property

P1:Reachability(D, d) t1 = 1

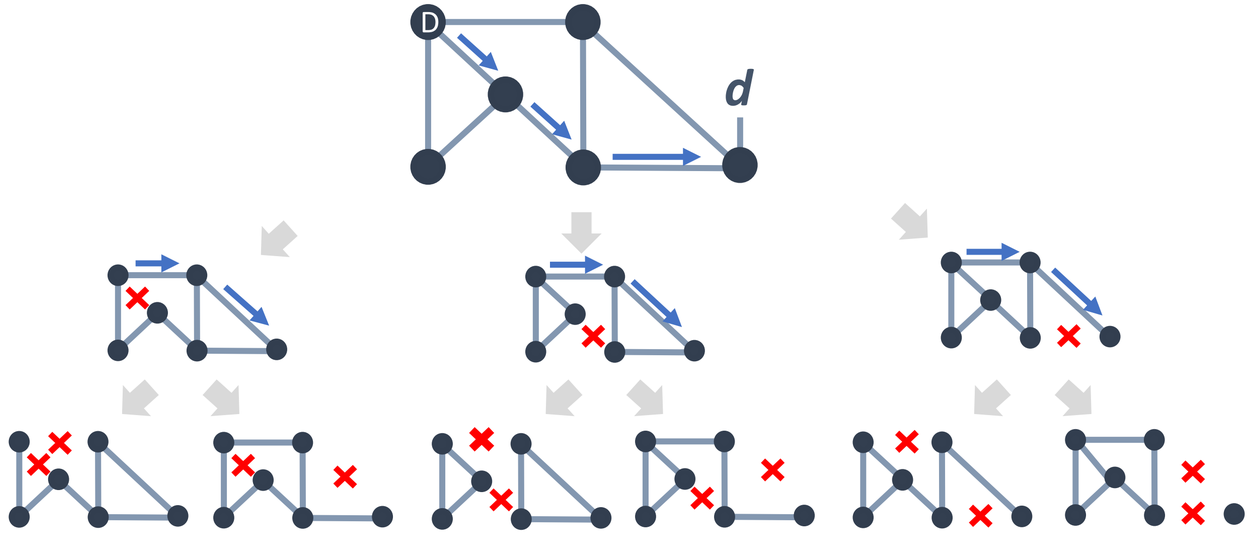
10 *failure scenarios*



Failure Scenarios for Single Property

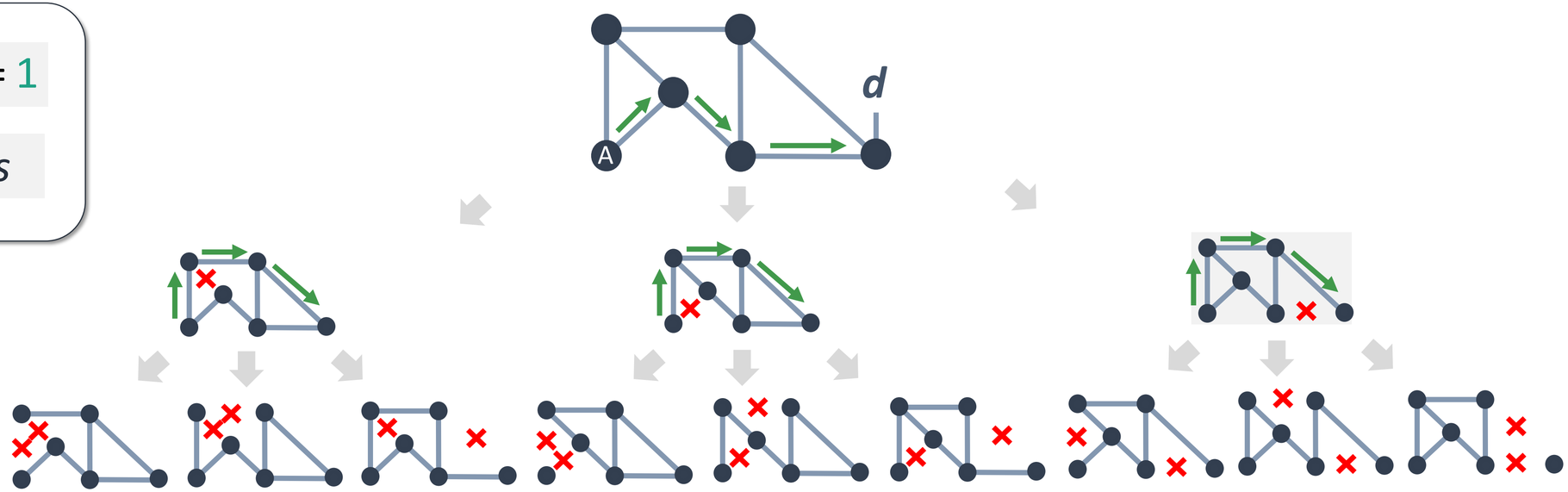
P1:Reachability(D, d) t1 = 1

10 *failure scenarios*



P2:Reachability(A, d) t2 = 1

13 *failure scenarios*



The number of Scenarios Remains Large

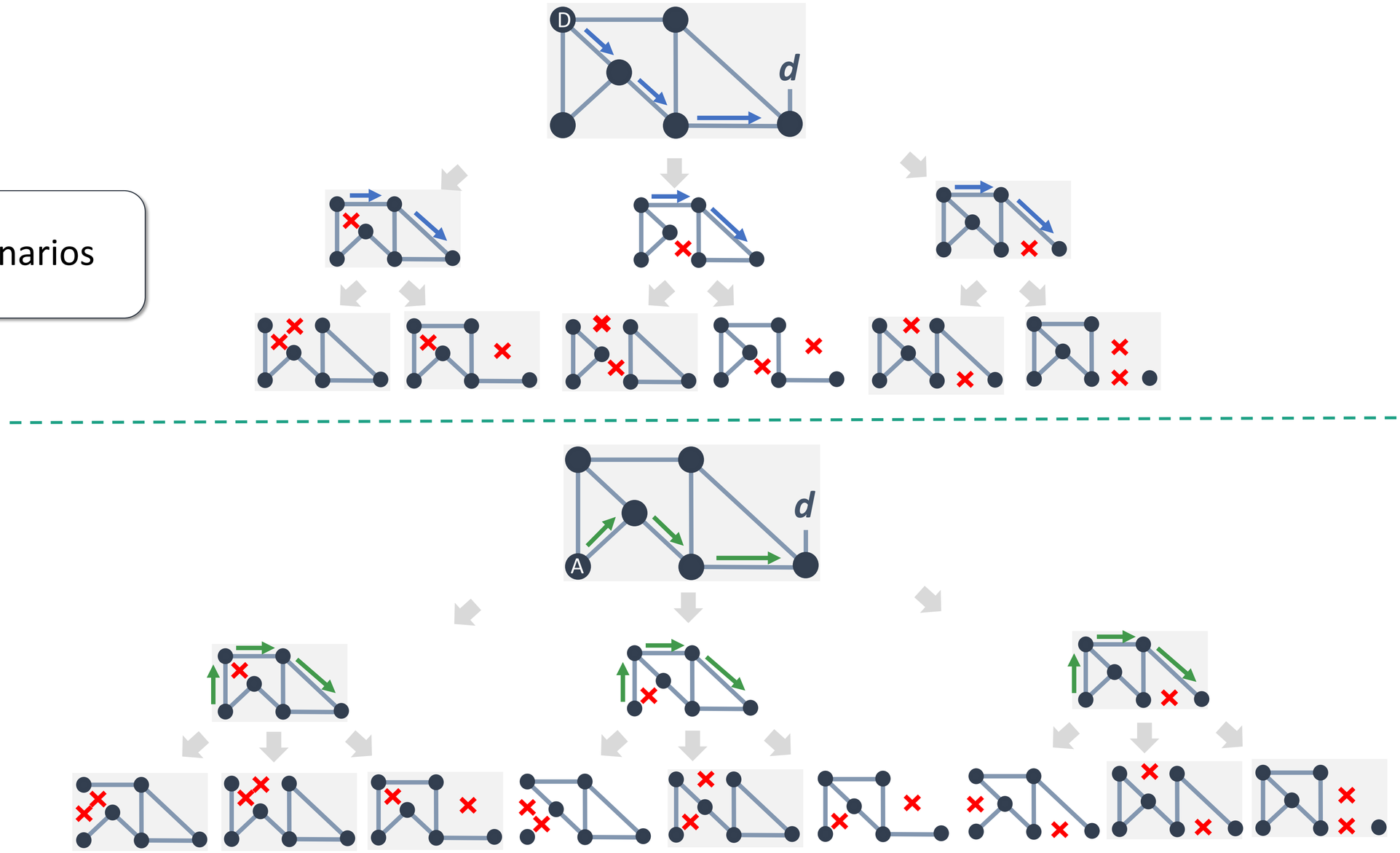
Recall : the number of properties is **large**

The number of scenarios selected by hot links is still large

	# of failure scenarios
DC1	10^6
DC2	10^4

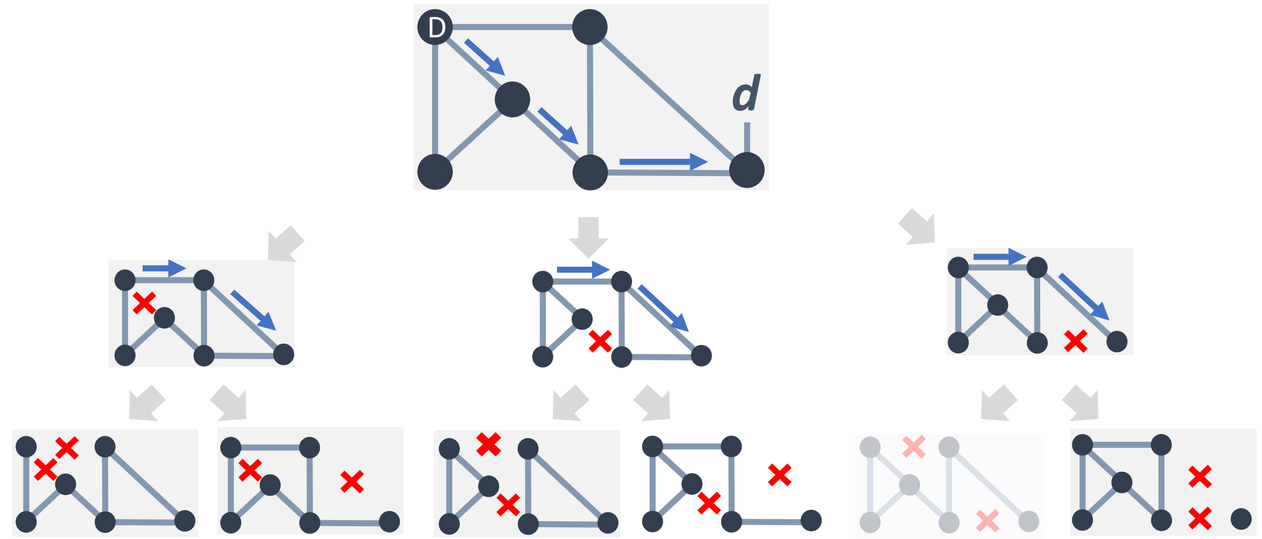
Aggregate Failure Scenarios

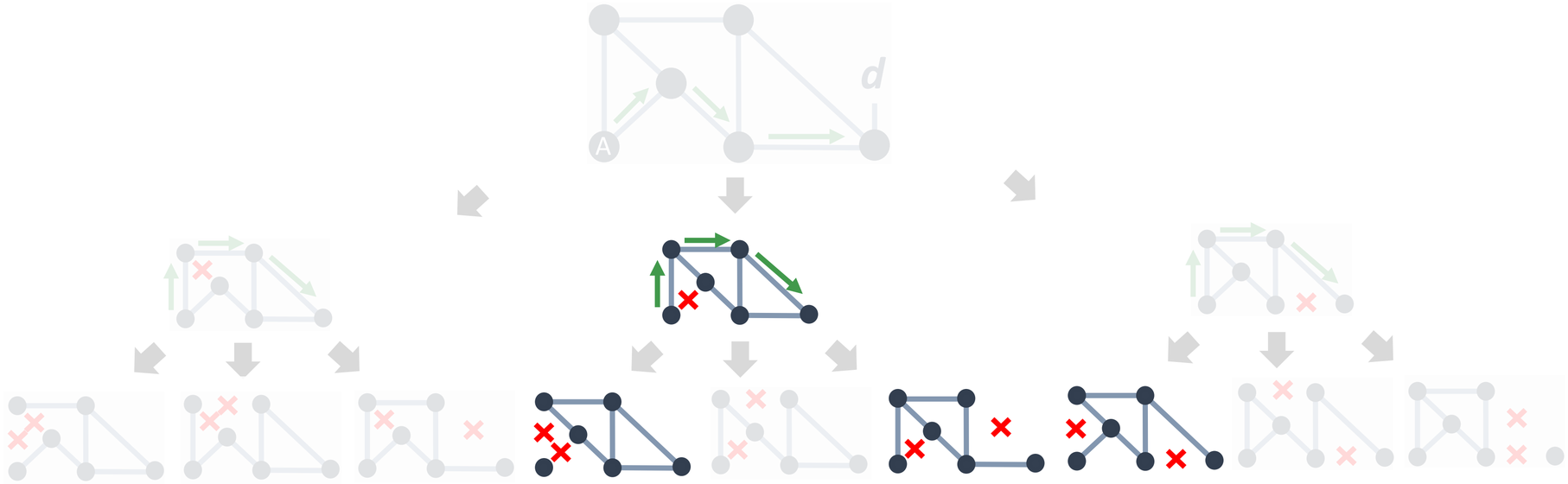
17 same failure scenarios



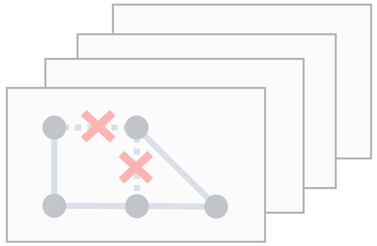
Aggregate Failure Scenarios

Save 10 failure scenarios



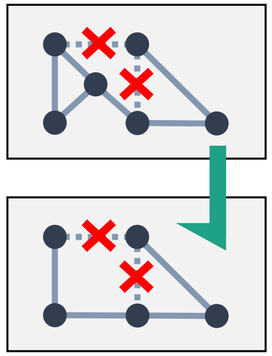


Challenge 2



- How to efficiently generate failure scenarios ?

  the number of failure scenarios

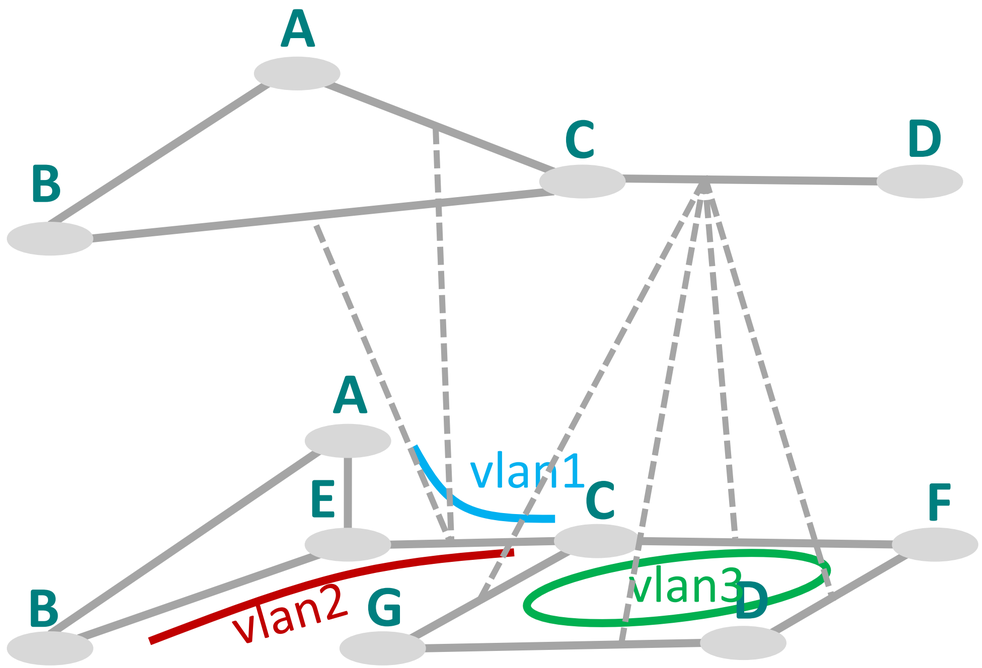


- How to make layer-3 topology generation scalable ?

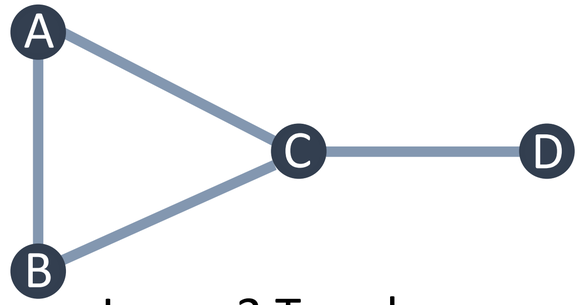
  the time of analyzing a single failure scenario

Fast Topology Mapping (FTM)

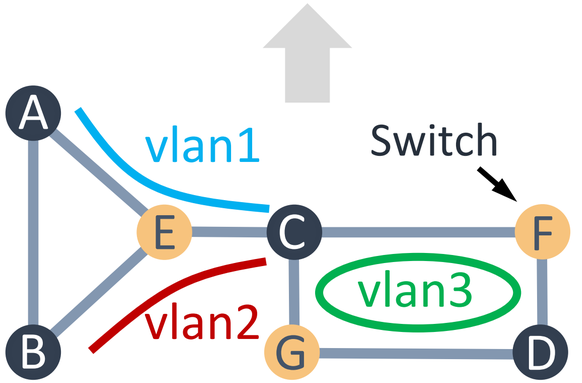
Fast Topology Mapping



Fast Topology Mapping

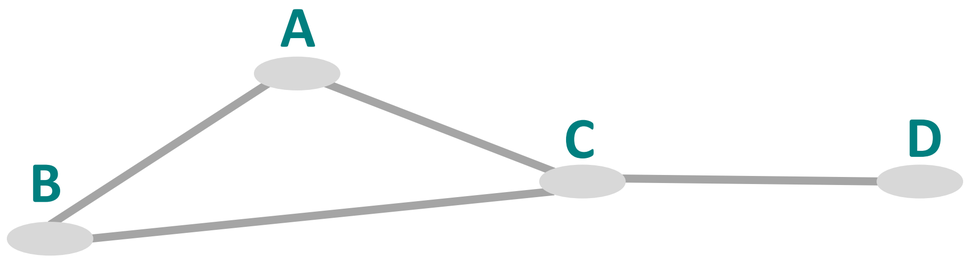


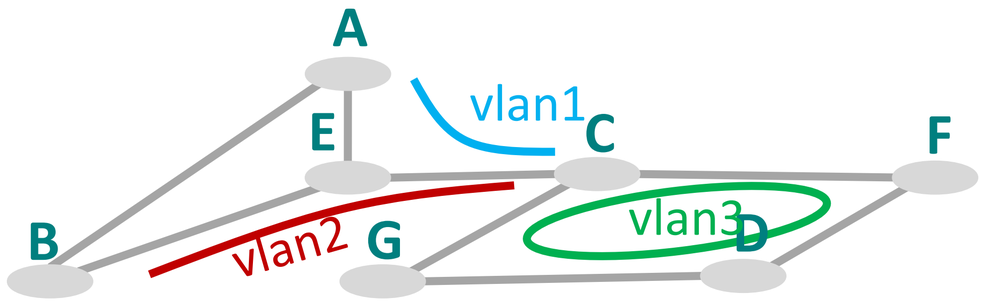
Layer-3 Topology



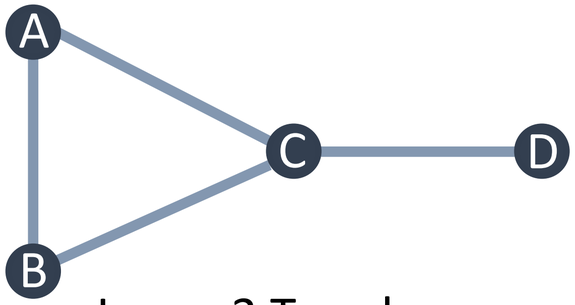
Layer-1 Topology

Running Example

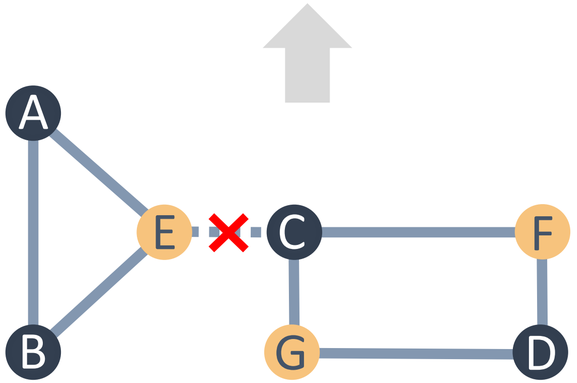




Fast Topology Mapping



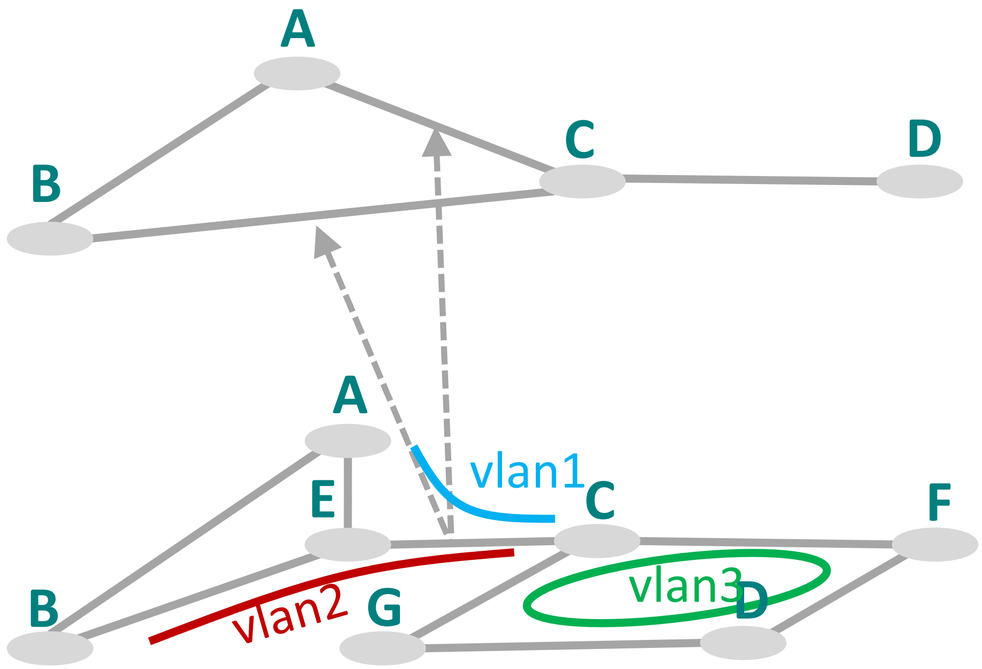
Layer-3 Topology



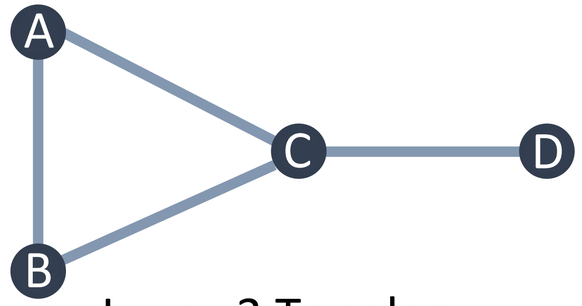
Layer-1 Topology

Fail layer-1 links

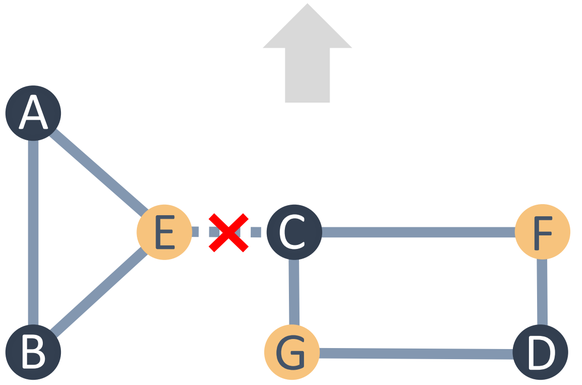
Running Example



Fast Topology Mapping



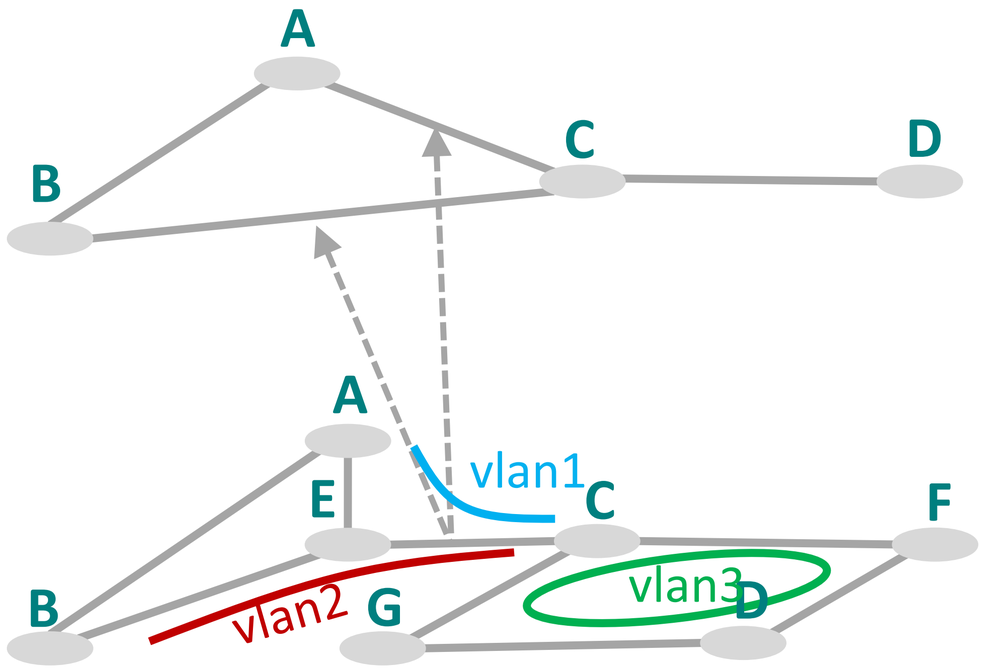
Layer-3 Topology



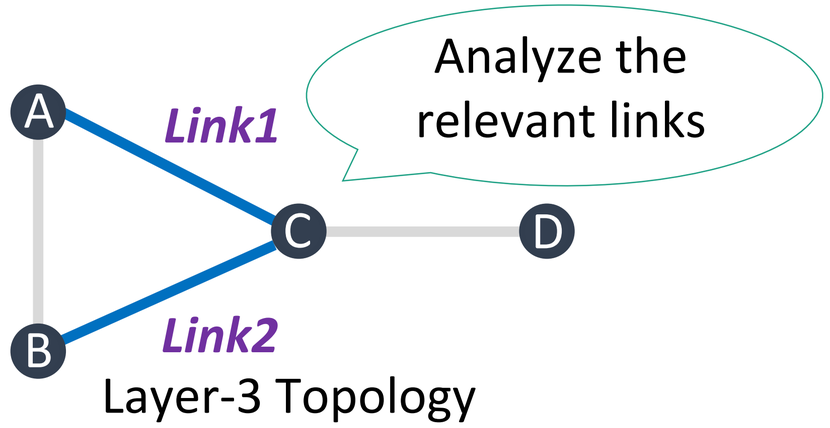
Layer-1 Topology

Fail layer-1 links → Get effected layer-3 links

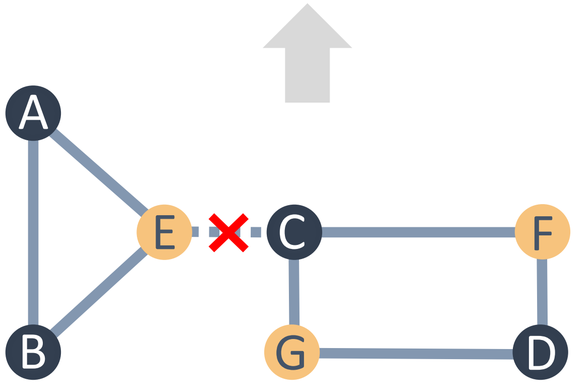
Running Example



Fast Topology Mapping



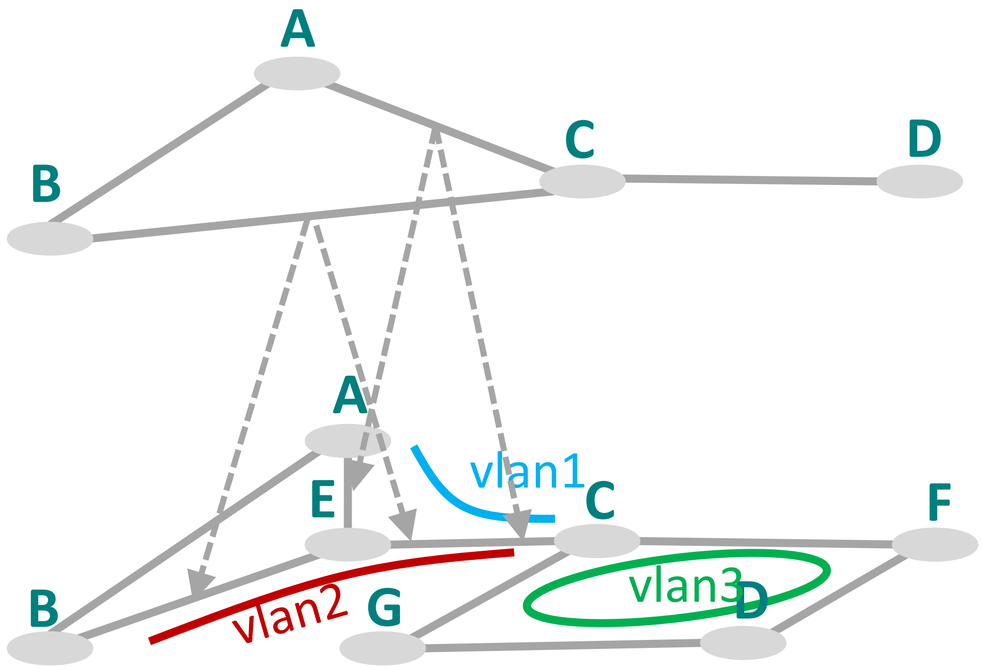
Layer-3 Topology



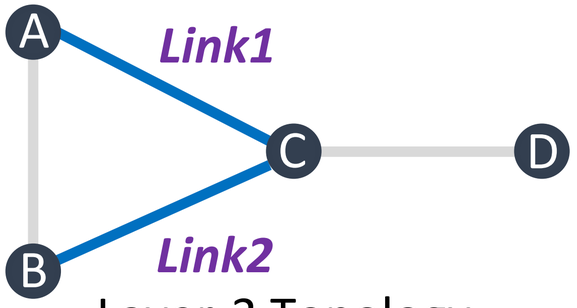
Layer-1 Topology

Fail layer-1 links → Get effected layer-3 links

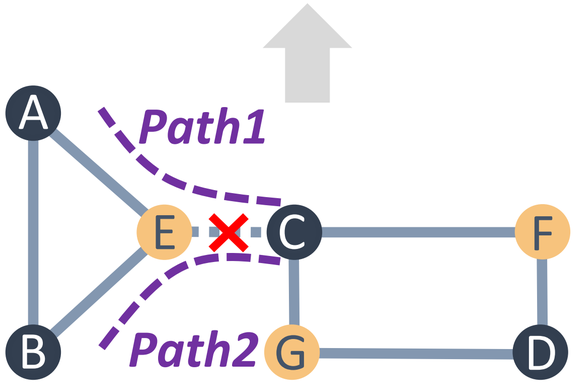
Running Example



Fast Topology Mapping



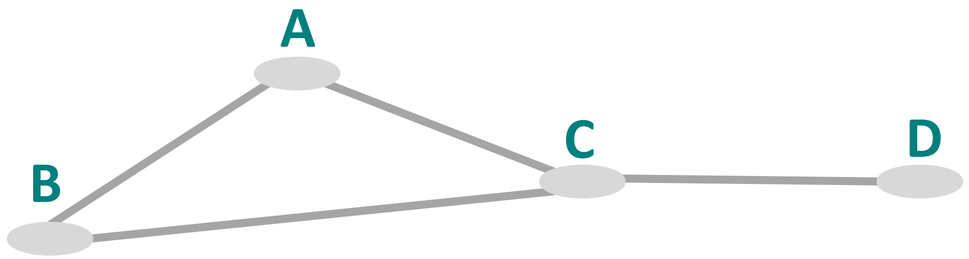
Layer-3 Topology

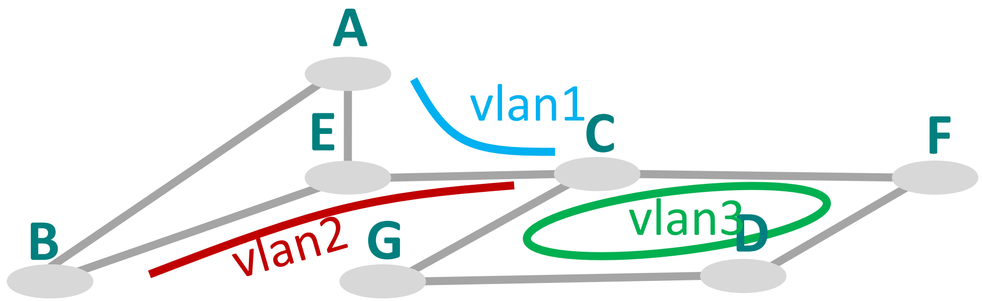


Layer-1 Topology

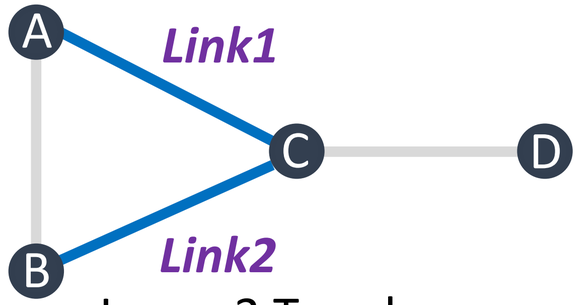
Fail layer-1 links → Get effected layer-3 links → Check layer-1 paths

Running Example

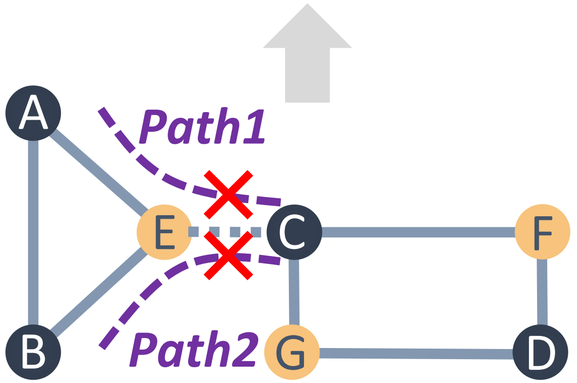




Fast Topology Mapping

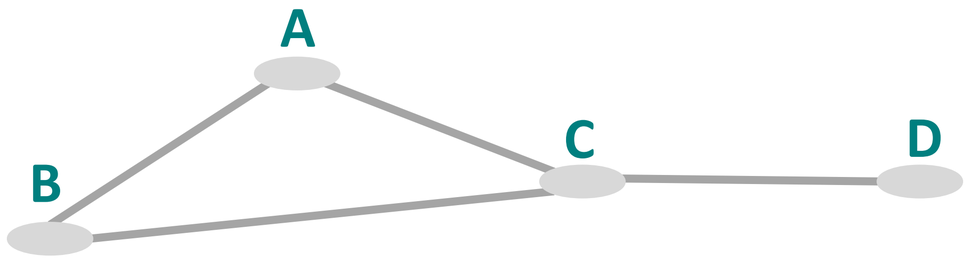


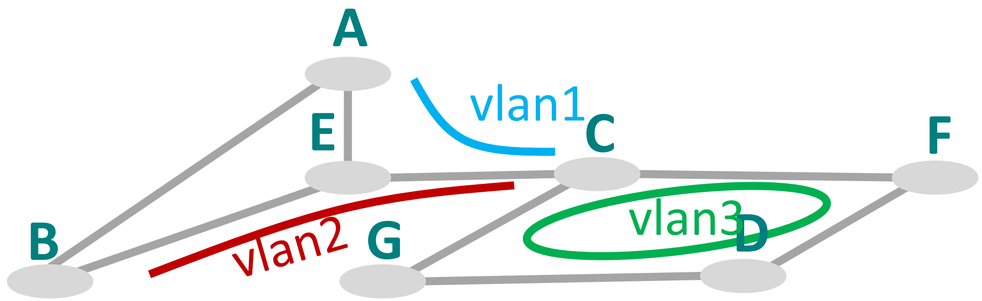
Layer-3 Topology



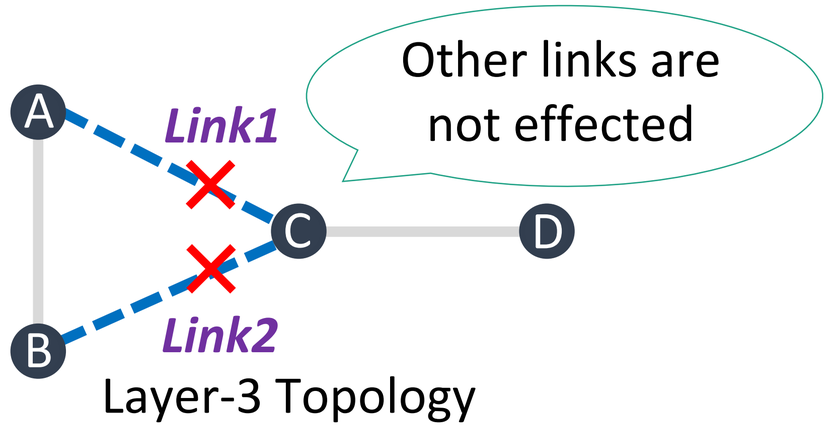
Fail layer-1 links → Get effected layer-3 links → Check layer-1 paths → Get layer3 failed links

Running Example

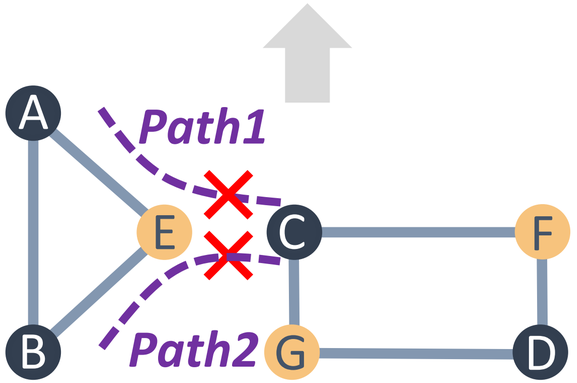




Fast Topology Mapping

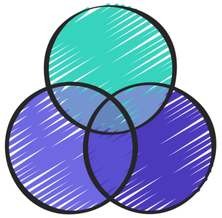


Layer-3 Topology



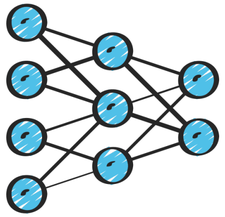
Fail layer-1 links → Get effected layer-3 links → Check layer-1 paths → Get layer3 failed links

Optimizations



1 Trimming based on enumeration analysis

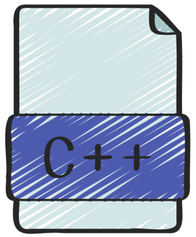
Enumeration when the number of failed links is small



2 Trimming based on topology condidtion

Filter properties that do not satisfy the minimum cut requirement

Evaluation



1 8k lines of C++ & 1k lines of Java

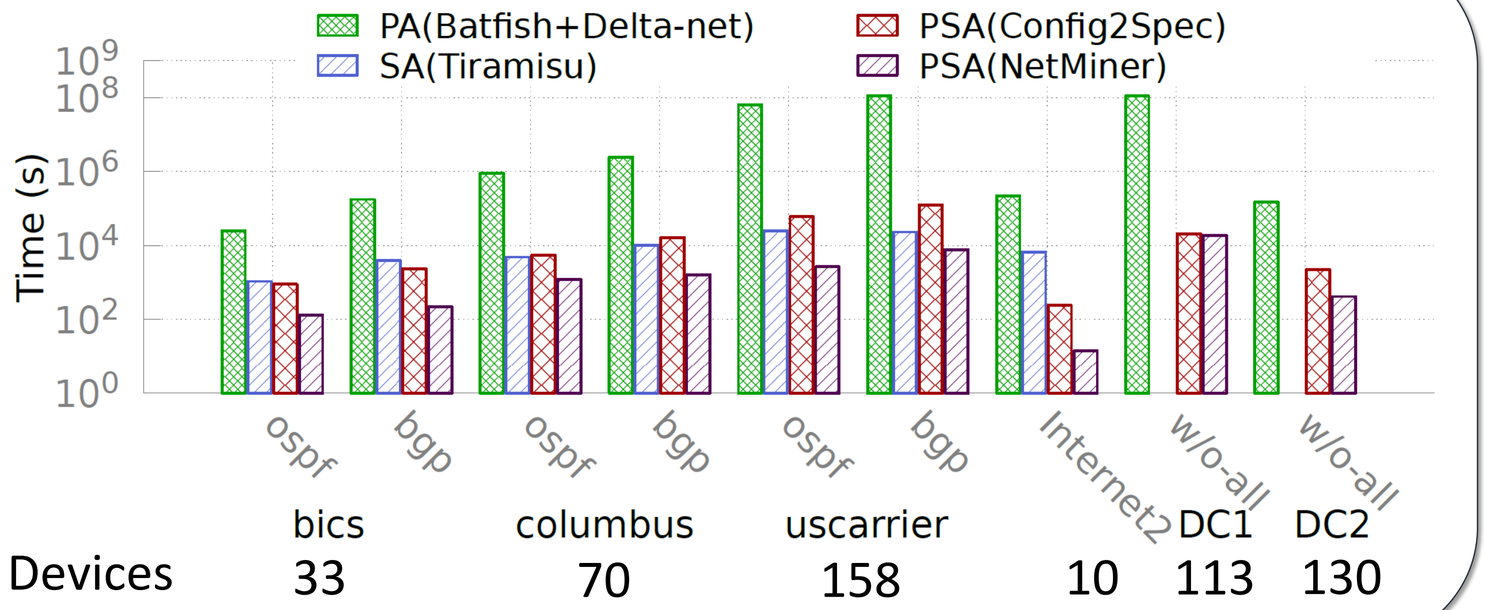
Relying on Batfish [NSDI` 15] and Delta-net [NSDI` 17]

2 Experiment on three real and three synthetic datasets

Real : 10 (Internet2), 113 and 130 (two data center networks) devices

Synthetic : with 33, 70, and 158 devices

Evaluation : Scalability



Faster than

Batfish+Delta-net : 10^2-10^5

Config2Spec : 6-16X

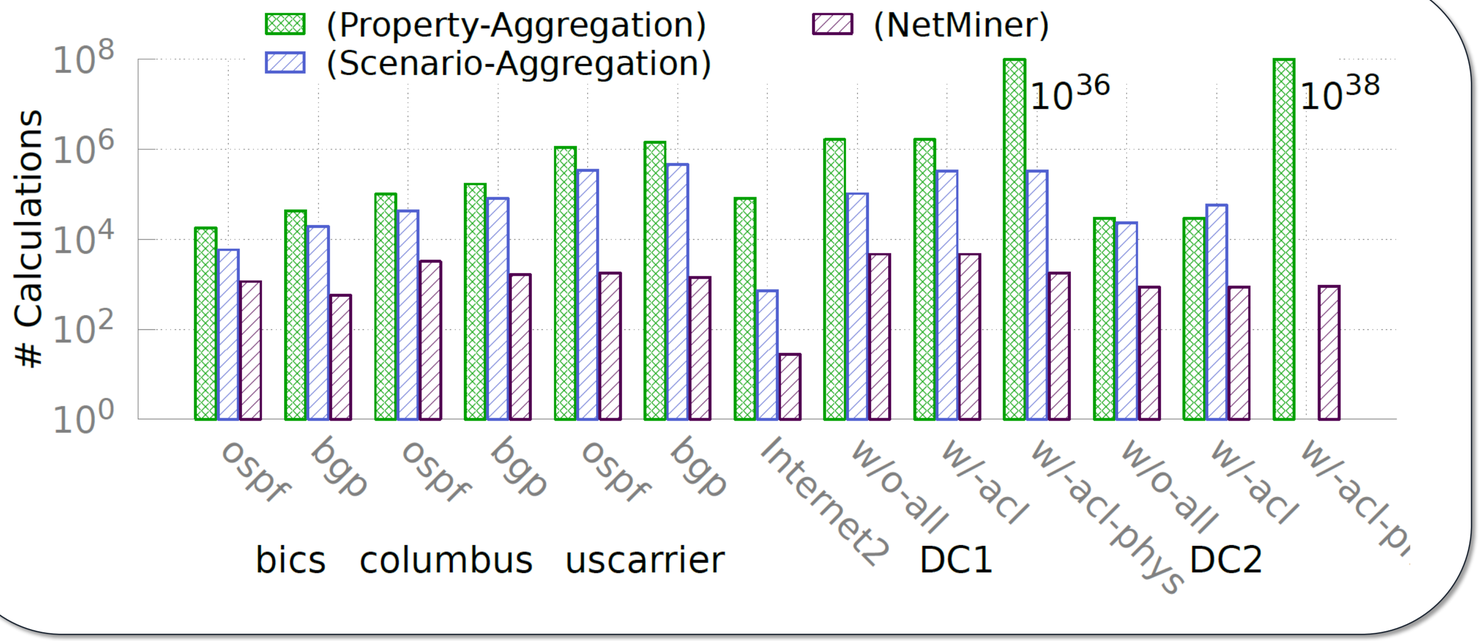
Tiramisu : 2-10X

Datasets	W/O ALL (s)	W/ ACL (s)	W/ ACL-PHYS (s)
DC1	18901	23004	16341
DC2	733	990	1083

Consider VRF, ACL, PHY Link, ...

No significant increase in run time

Evaluation : Microbenchmark



GSA reduces # of Scenarios than

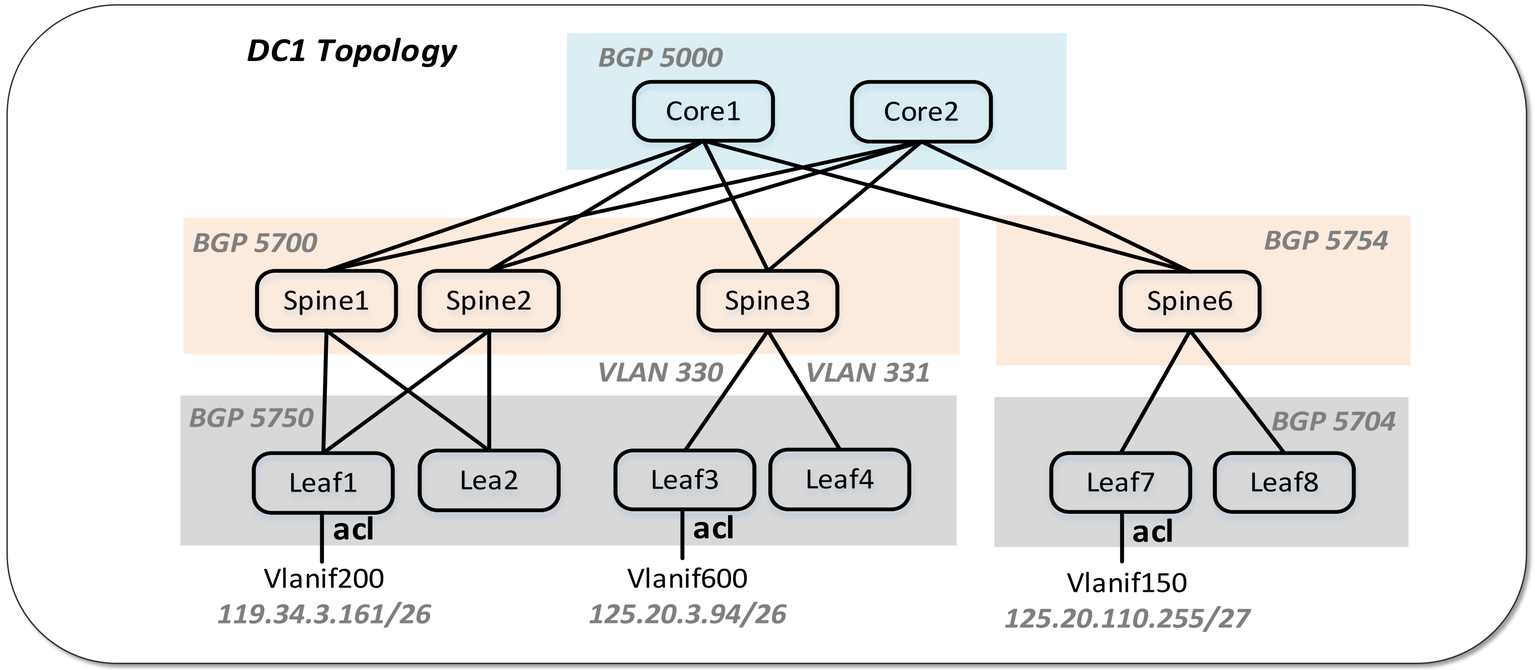
Enumeration(Batfish) : 10^2 to 10^3

Datasets	Layer-3 Topology Computation Time Batfish l=1,2,3	NetMiner l=1	NetMiner l=2	NetMiner l=3	Simulation Time Batfish -
DC1	7.8s	9us	11us	15us	3.1s
DC2	192.7s	676us	1076us	1423us	0.54s

FTM fast than

Batfish : 10^5 to 10^6

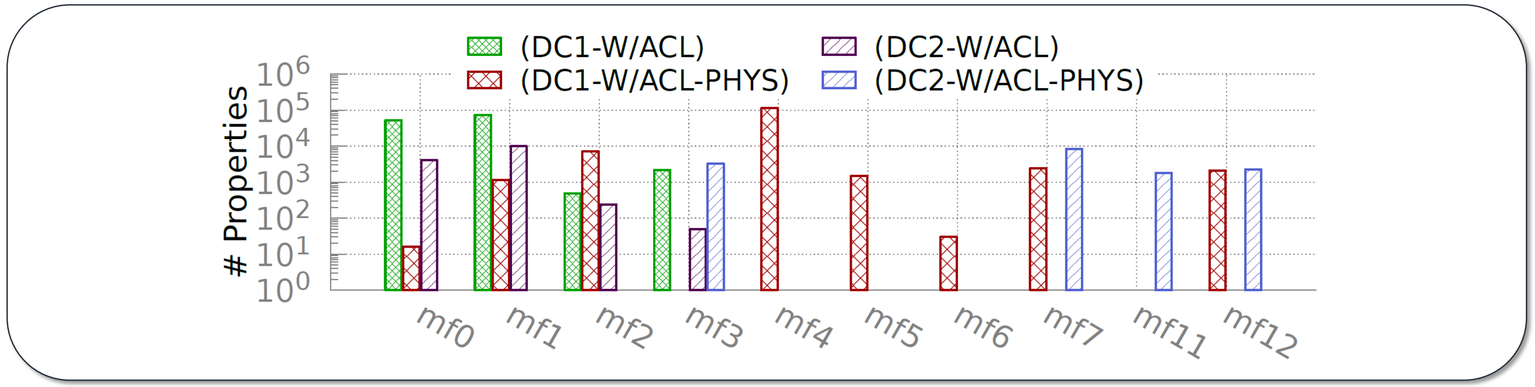
Evaluation : Fidelity



Applying ACL

Reachability from 127079→126848

NetMiner can avoid these faults



Applying Vlan & PHY Links

The tolerance distribution is larger

Conclusions

NetMiner is a general and scalable specification mining tool

- Using ***pure simulation*** to achieve high ***fidelity***

- Using ***general scenarios aggregation*** and ***fast topology mapping*** to achieve high ***scalability***

Questions

NetMiner