

# Probabilistic Verification of Network Configurations



Samuel  
Steffen



Timon  
Gehr



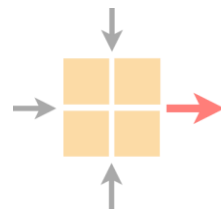
Petar  
Tsankov



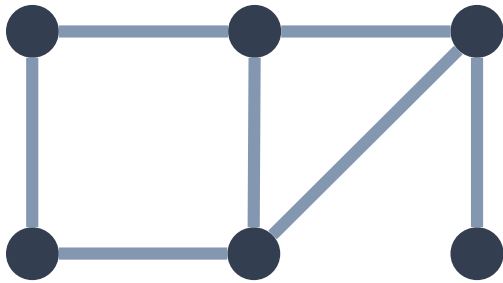
Laurent  
Vanbever



Martin  
Vechev

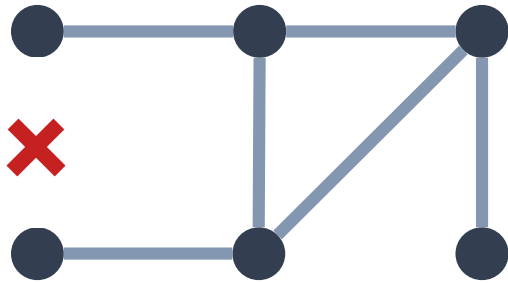


# Traditional Verification



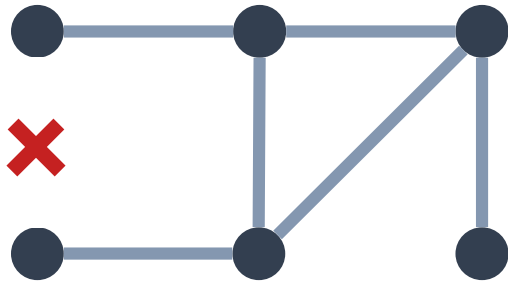
Reachable?

# Traditional Verification



Reachable?

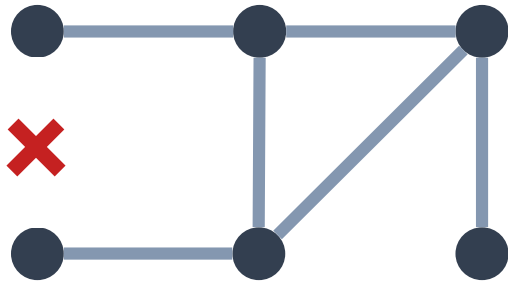
# Traditional Verification



Reachable?

Batfish, Minesweeper, ERA,  
ARC, Plankton, Tiramisu, ...

# Traditional Verification



Reachable?

Batfish, Minesweeper, ERA,  
ARC, Plankton, Tiramisu, ...

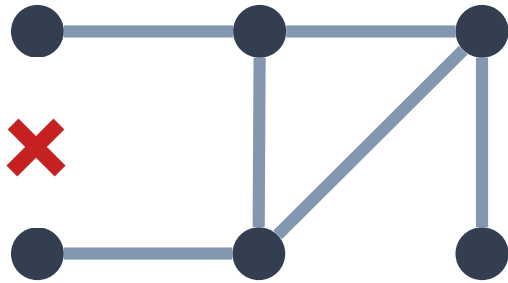


for a *specific* failure scenario



for *all* scenarios up to k failures

# Traditional Verification



Reachable?

“hard” property

Batfish, Minesweeper, ERA,  
ARC, Plankton, Tiramisu, ...

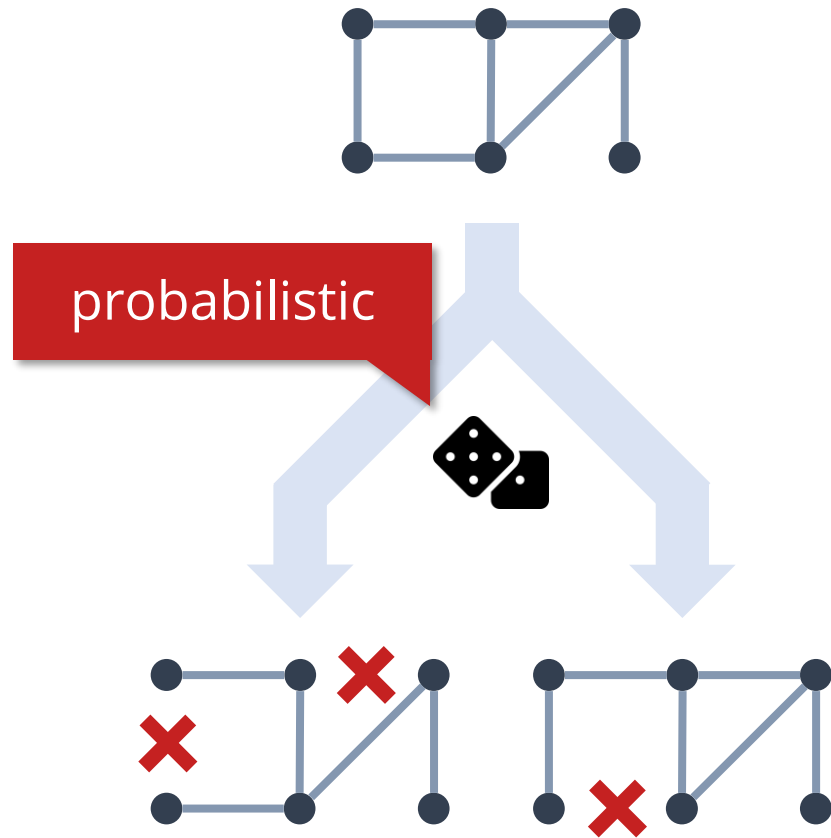


for a *specific* failure scenario

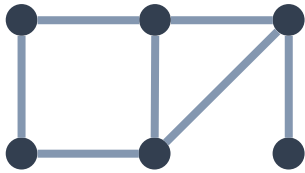


for *all* scenarios up to k failures

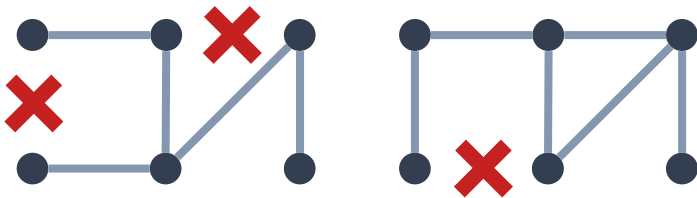
# ***Probabilistic Verification***



# Probabilistic Verification



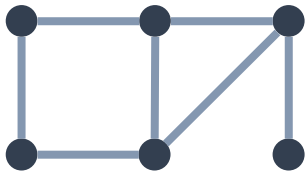
probabilistic



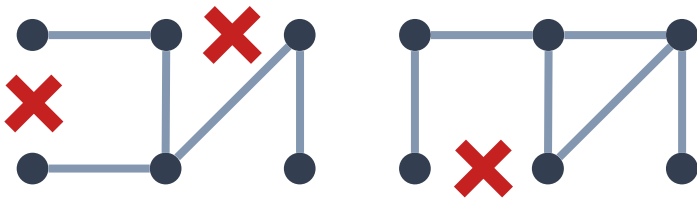
What is the *probability* of ?



# Probabilistic Verification



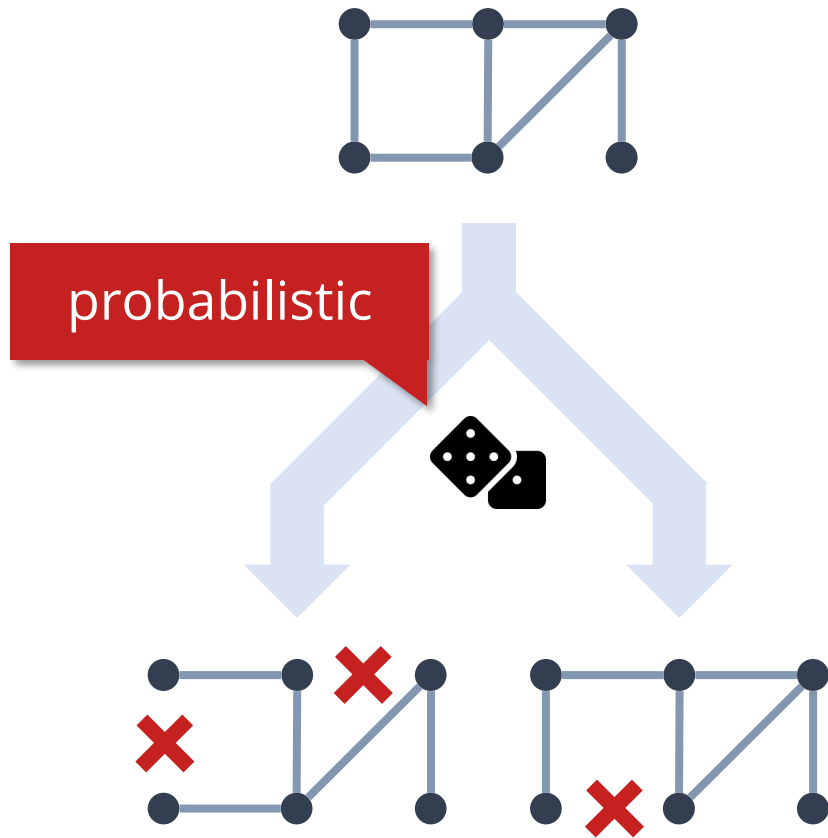
probabilistic



What is the *probability* of ?

Service Level Agreements (SLA)  
"99.99% *reachability*"

# Probabilistic Verification

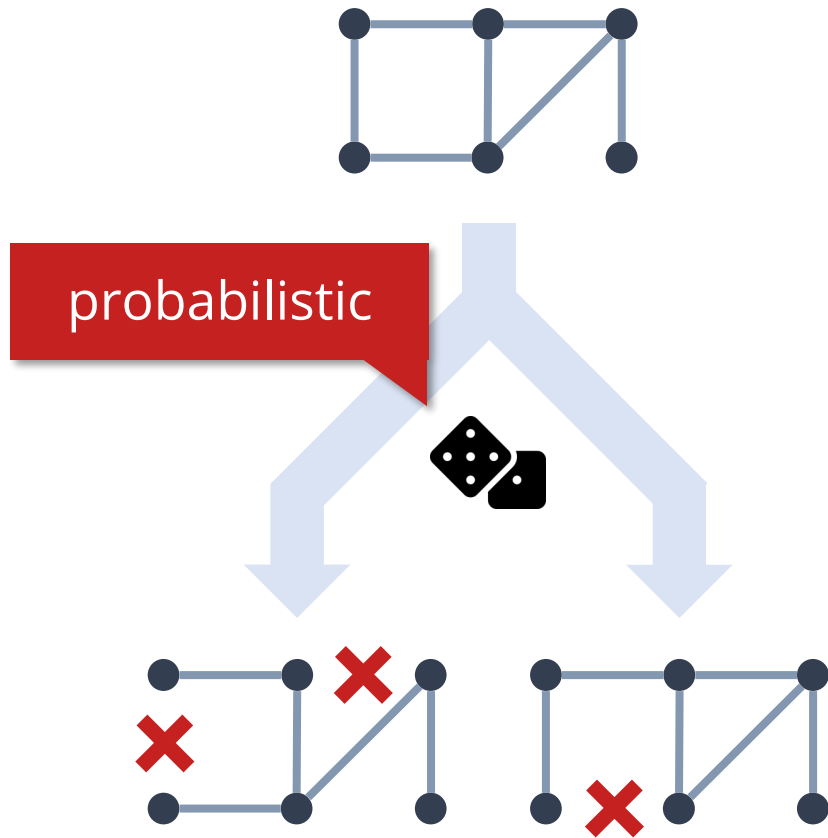


What is the *probability* of ?

Service Level Agreements (SLA)  
"99.99% *reachability*"

Traffic Engineering  
"80% *load-balanced*"

# Probabilistic Verification



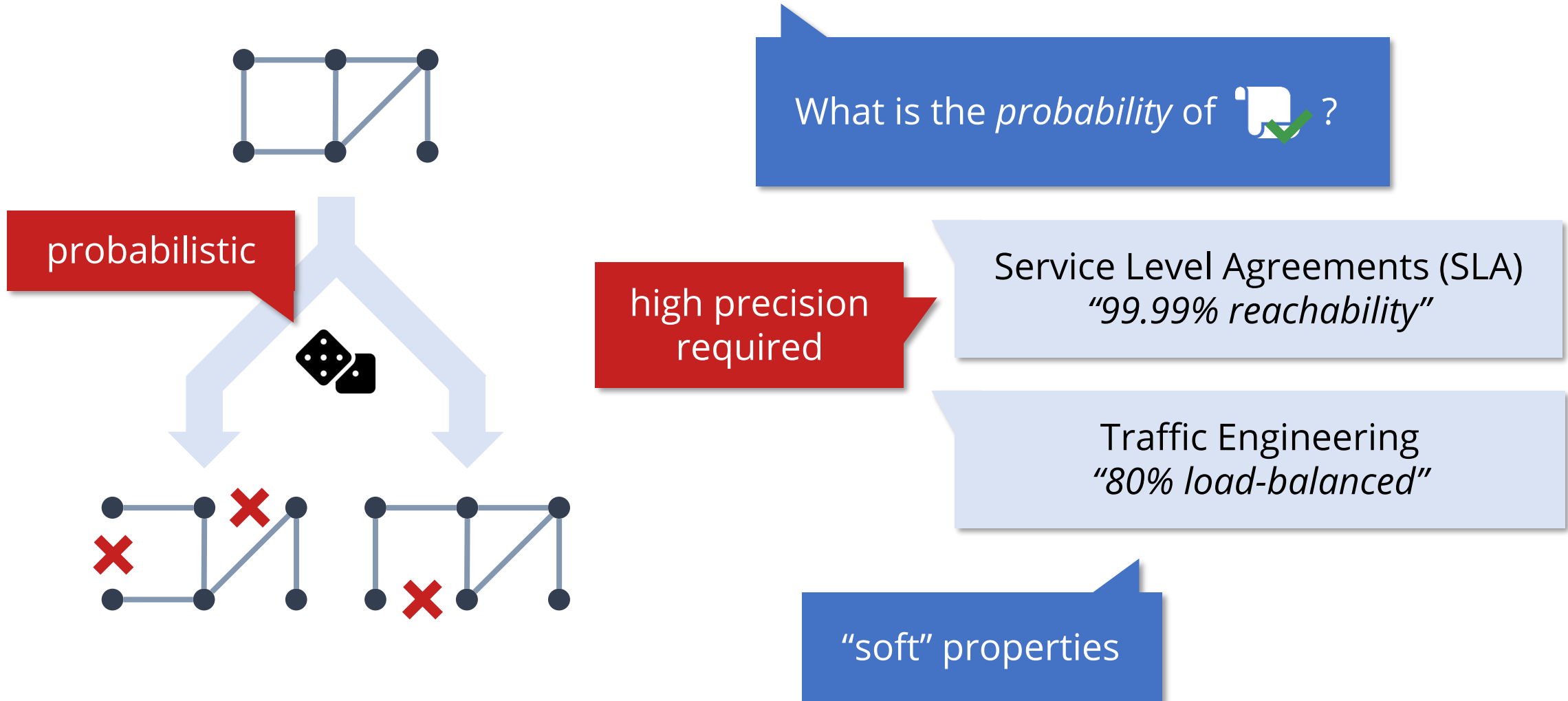
What is the *probability* of ?

Service Level Agreements (SLA)  
"99.99% *reachability*"

Traffic Engineering  
"80% *load-balanced*"

"soft" properties

# Probabilistic Verification



# Existing Work

TEAVAR [SIGCOMM 19]  
Lancet [SIGMETRICS 20]

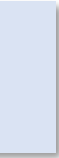
*Synthesis* for traffic engineering  
under probabilistic failures

# Existing Work

TEAVAR [SIGCOMM 19]  
Lancet [SIGMETRICS 20]

*Synthesis* for traffic engineering  
under probabilistic failures

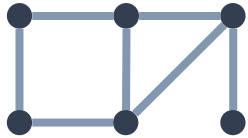
No support for BGP or OSPF



# Attempt I: Partial Exploration

# Attempt I: Partial Exploration

No failures

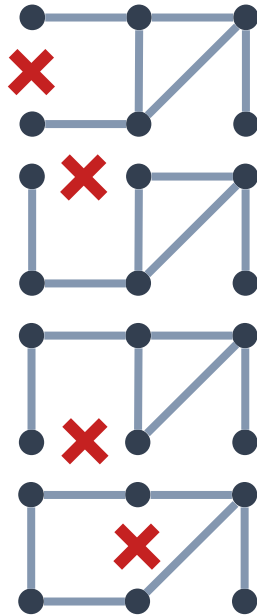
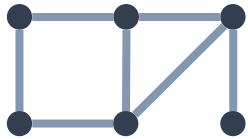




# Attempt I: Partial Exploration

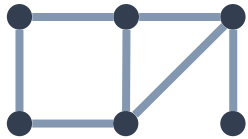
1 link failure

No failures

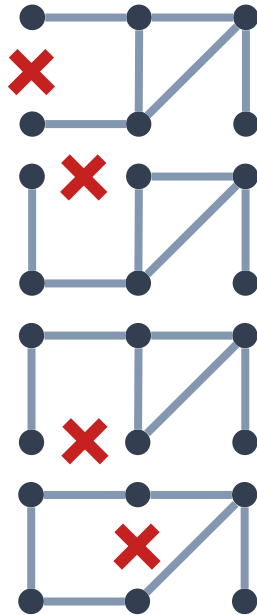


# Attempt I: Partial Exploration

No failures

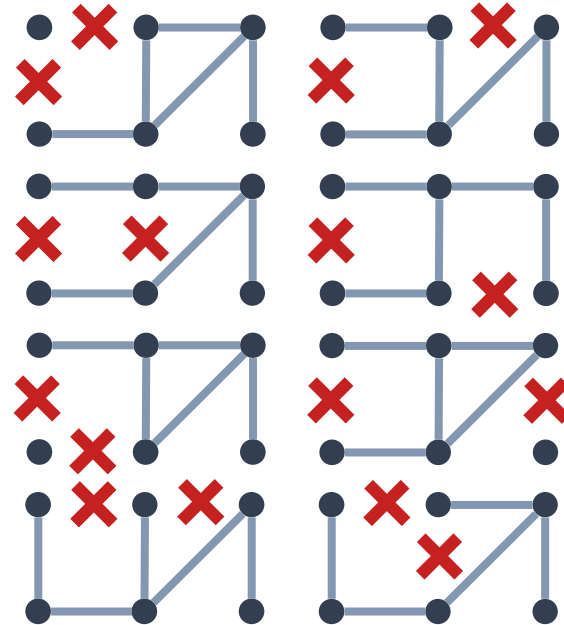


1 link failure



...

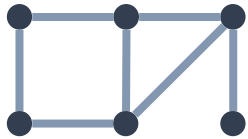
2 link failures



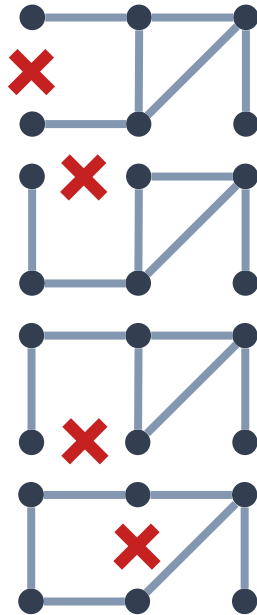
...

# Attempt I: Partial Exploration

No failures

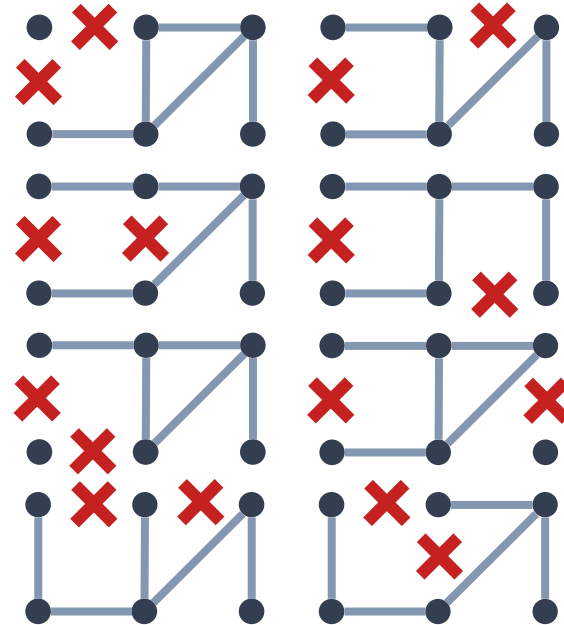


1 link failure



...

2 link failures

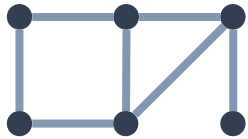


...

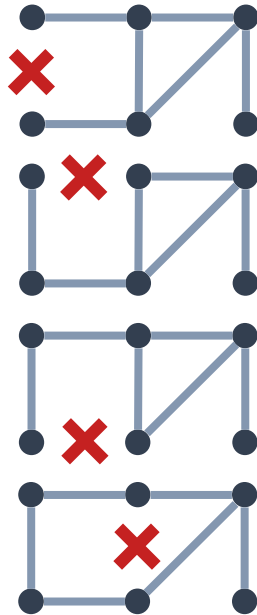
...

# Attempt I: Partial Exploration

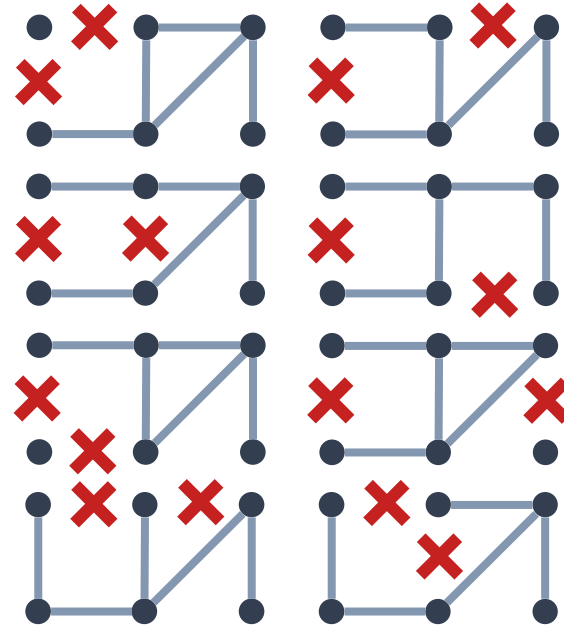
No failures



1 link failure



2 link failures



...

Exponentially less likely scenarios

# Attempts

Partial exploration

#scenarios for *four* 9s,  
191 links,  $p_{\text{link failure}} = 0.001$

# Attempts

Partial exploration

**1 107 359**

#scenarios for *four* 9s,  
191 links,  $p_{\text{link failure}} = 0.001$

# Attempts

Too expensive

Partial exploration

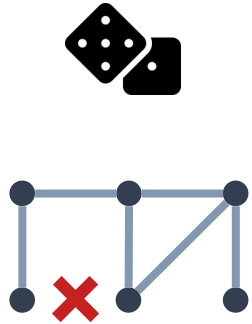
**1 107 359**

#scenarios for *four* 9s,  
191 links,  $p_{\text{link failure}} = 0.001$

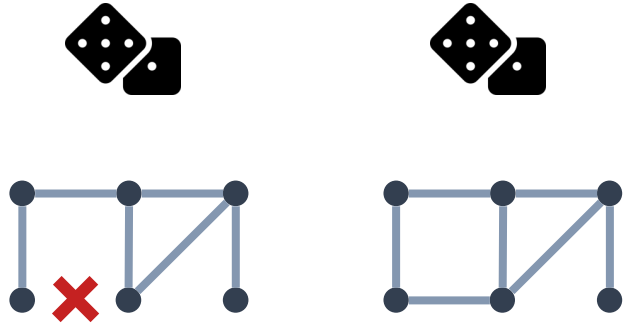
# Attempt II: Estimation via Sampling



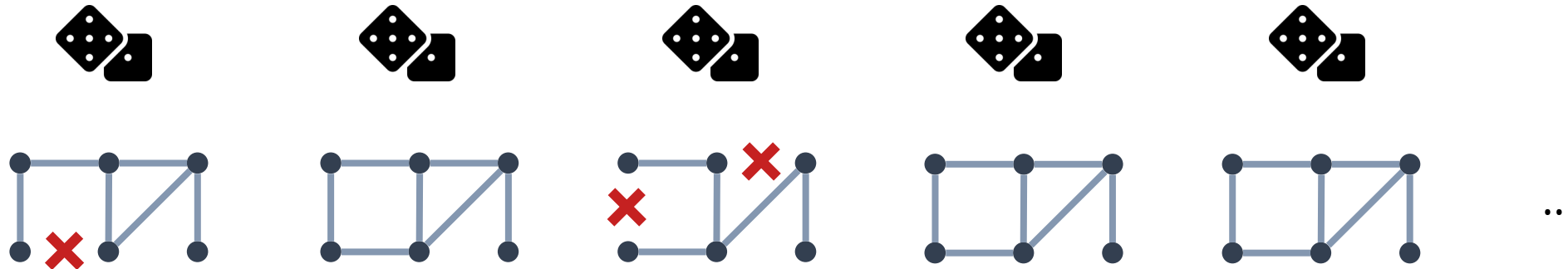
# Attempt II: Estimation via Sampling



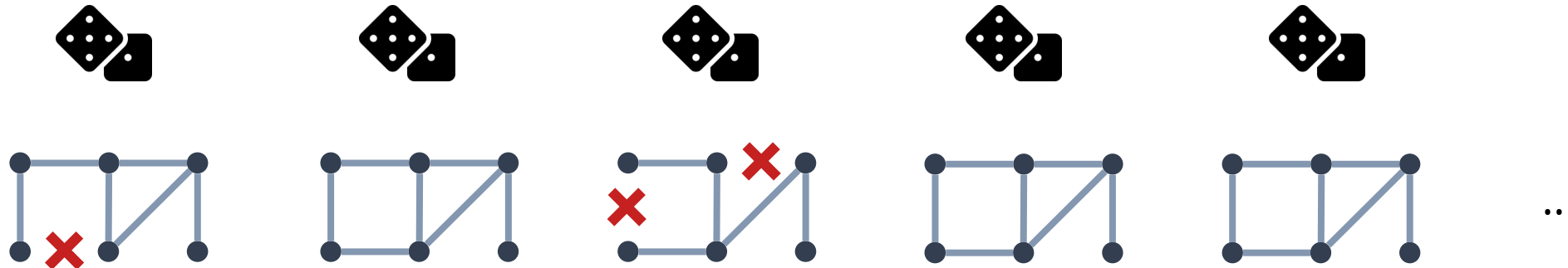
# Attempt II: Estimation via Sampling




# Attempt II: Estimation via Sampling



# Attempt II: Estimation via Sampling



Compute fraction of samples where 

# Attempts

Too expensive

Partial exploration

Estimation via  
sampling

**1 107 359**

#scenarios for *four* 9s,  
191 links,  $p_{\text{link failure}} = 0.001$

# Attempts

Too expensive

Partial exploration

**1 107 359**

#scenarios for *four* 9s,  
191 links,  $p_{\text{link failure}} = 0.001$

Estimation via  
sampling

**738 M**

Hoeffding,  $\alpha = 0.95$

# Attempts

Too expensive

Partial exploration

**1 107 359**

#scenarios for *four* 9s,  
191 links,  $p_{\text{link failure}} = 0.001$

Estimation via  
sampling

**738 M**

Hoeffding,  $\alpha = 0.95$

# Attempts

Too expensive

Partial exploration

**1 107 359**

#scenarios for *four 9s*,  
191 links,  $p_{\text{link failure}} = 0.001$

Estimation via  
sampling

**738 M**

Hoeffding,  $\alpha = 0.95$



**1 854**



# Attempts

Too expensive

Partial exploration

**1 107 359**

#scenarios for *four 9s*,  
191 links,  $p_{\text{link failure}} = 0.001$

Estimation via  
sampling

**738 M**

Hoeffding,  $\alpha = 0.95$



**1 854**

≈600x reduction

# Overview



BGP + IGP support ✓

High accuracy 

Scalable 

# NetDice Overview



# NetDice Overview

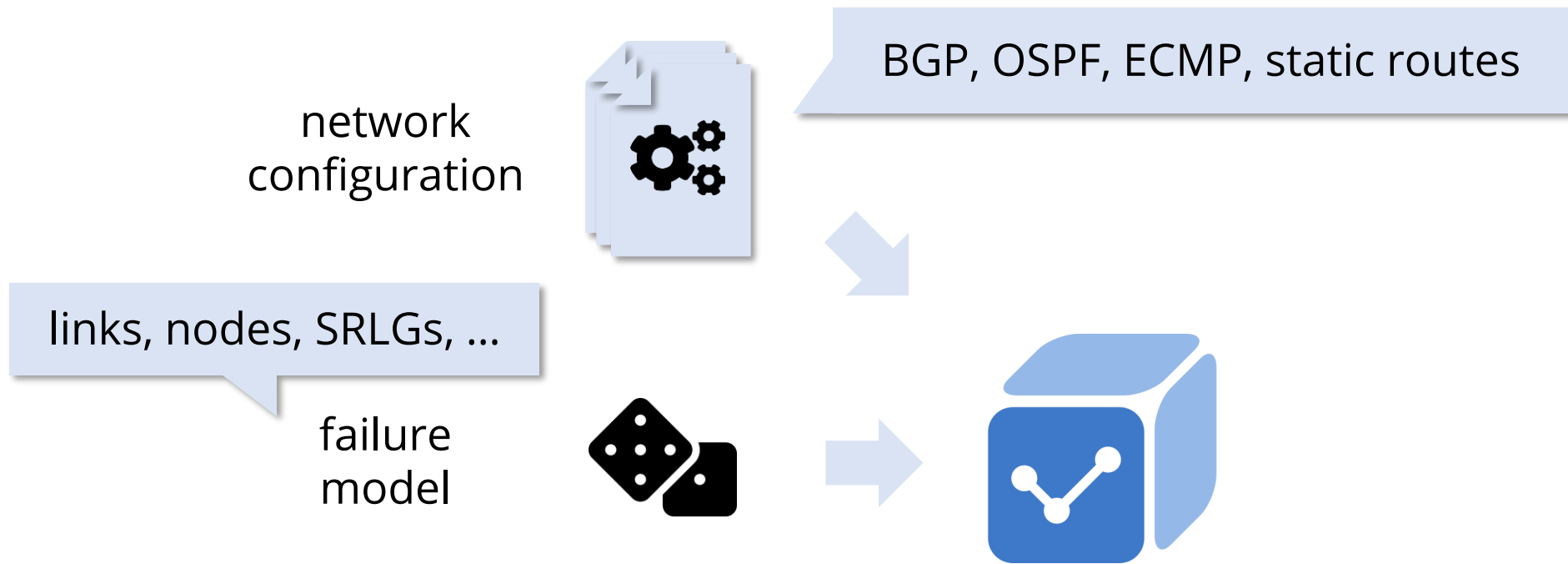
network  
configuration



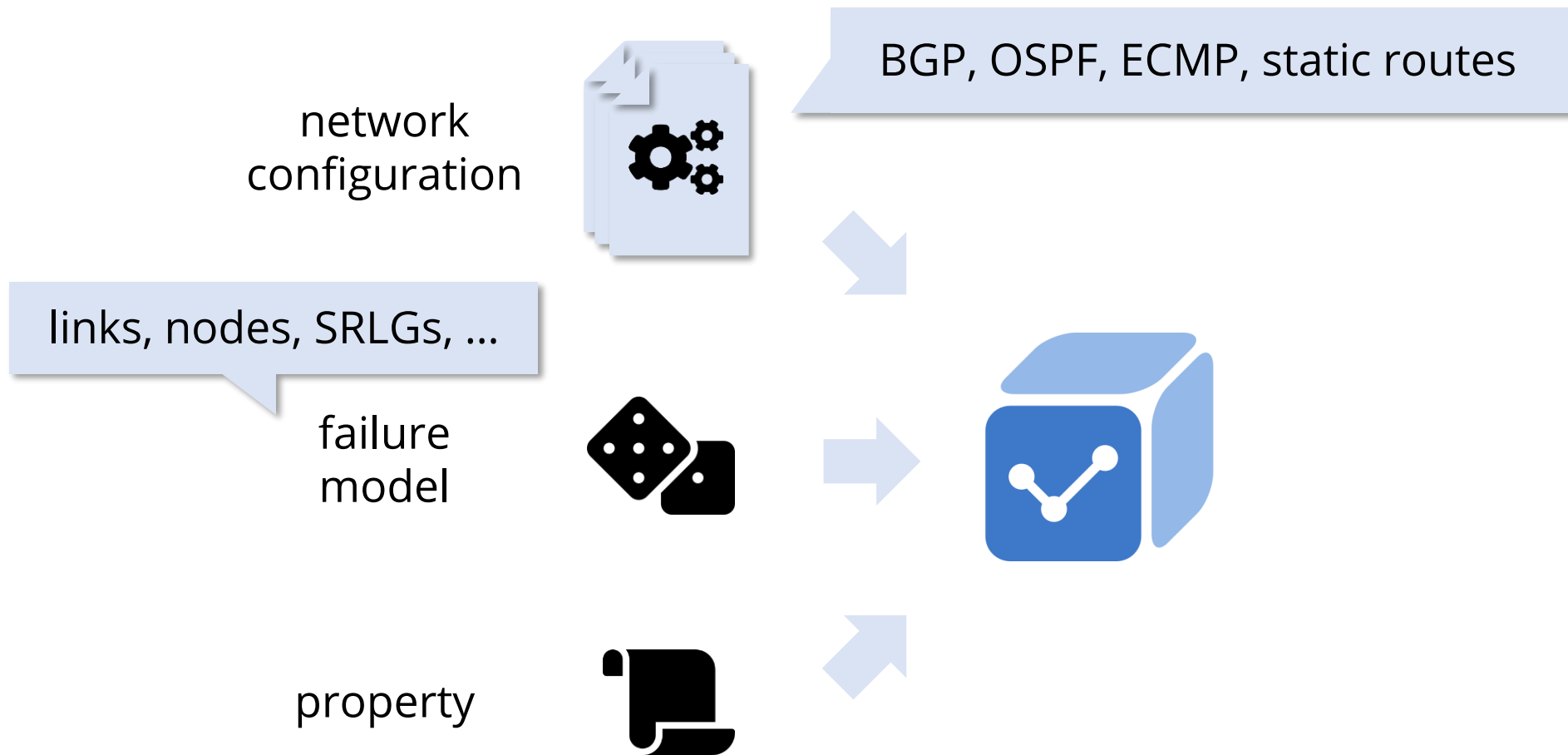
BGP, OSPF, ECMP, static routes



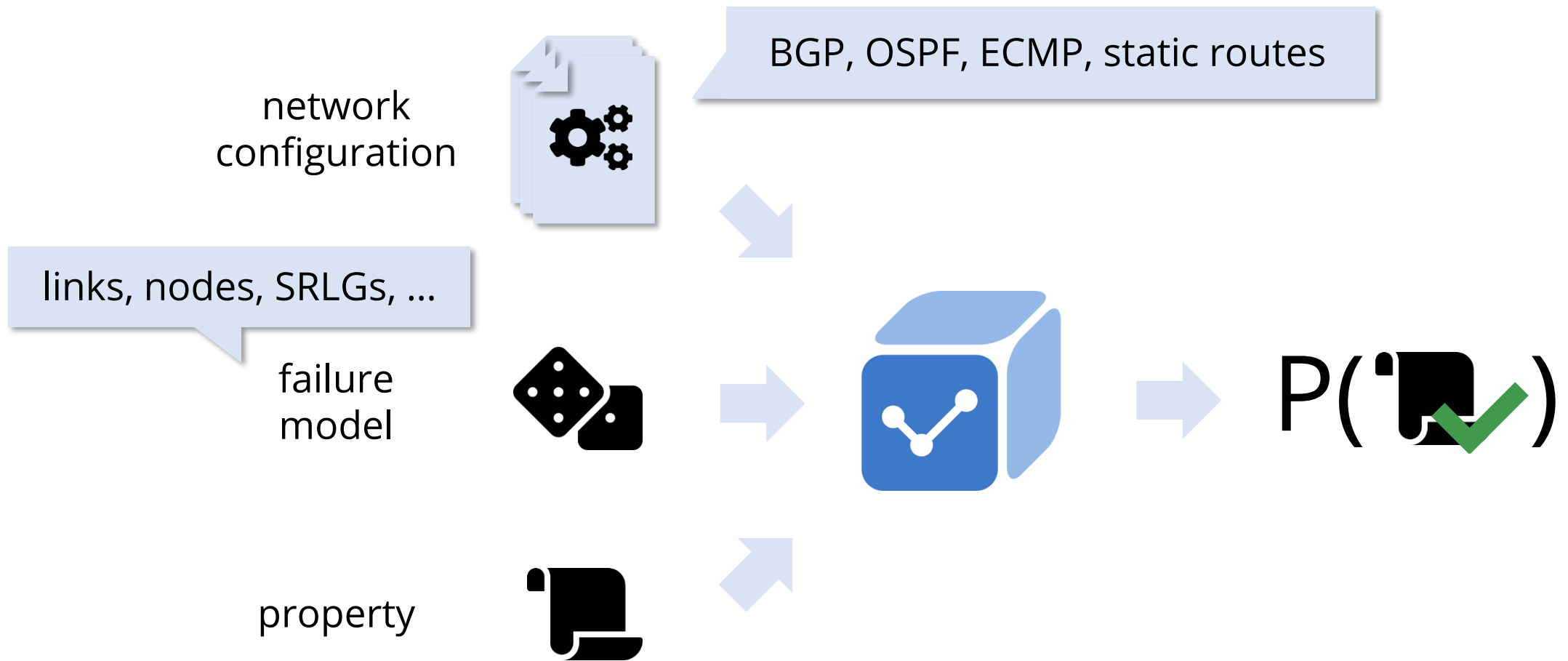
# NetDice Overview



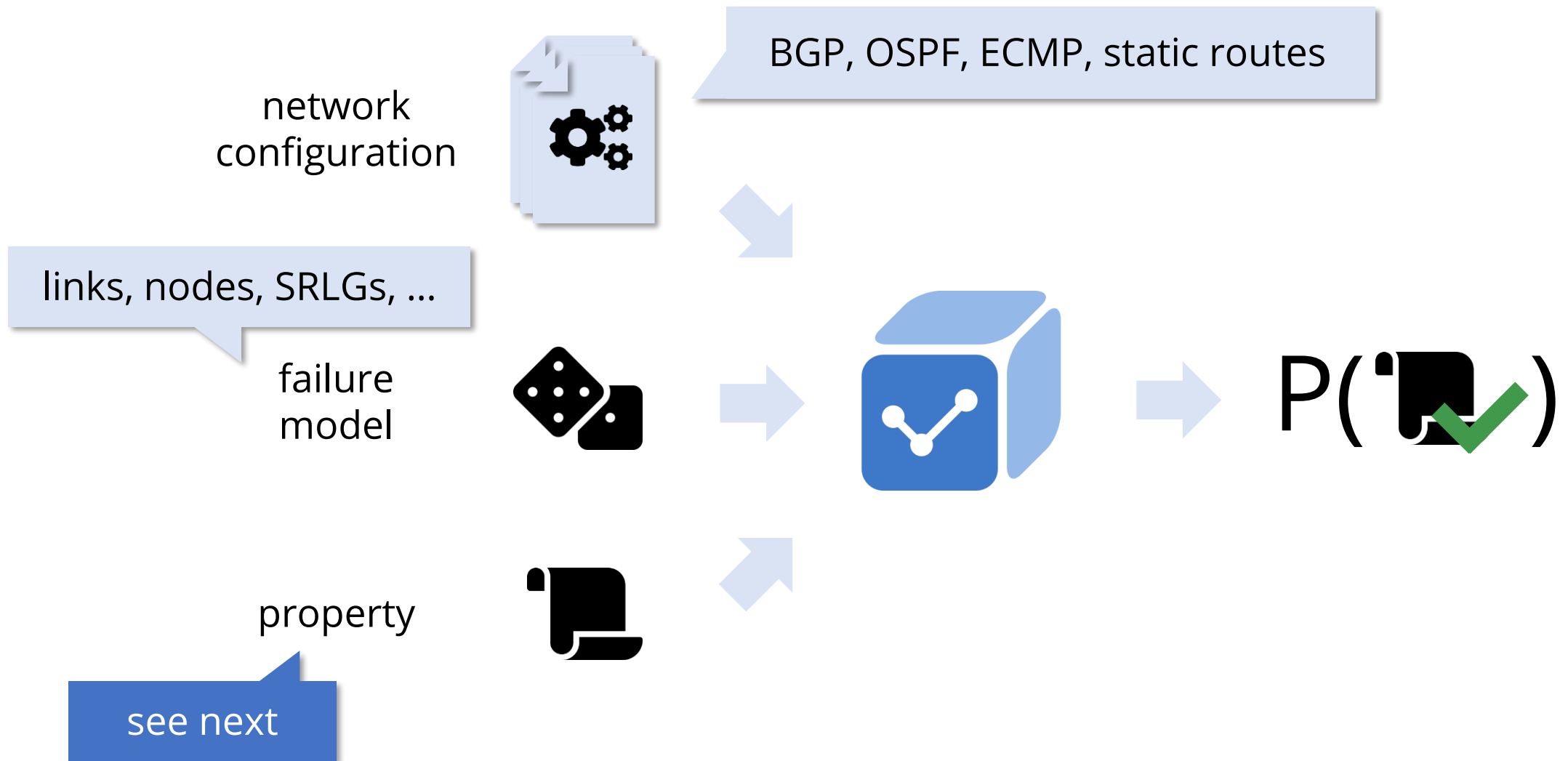
# NetDice Overview



# NetDice Overview

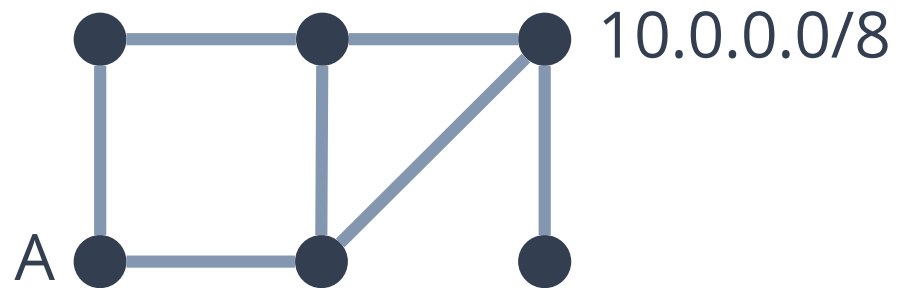


# NetDice Overview



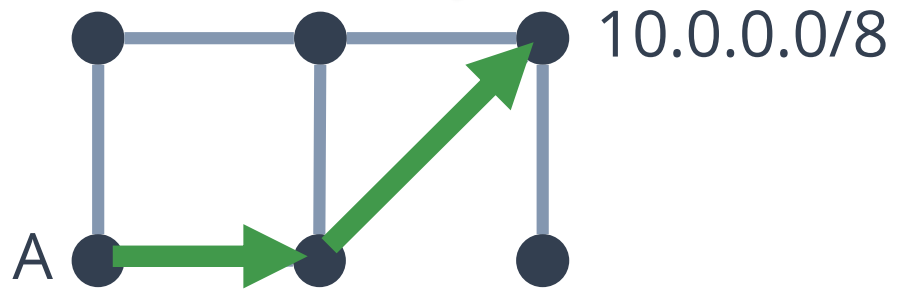


# Properties



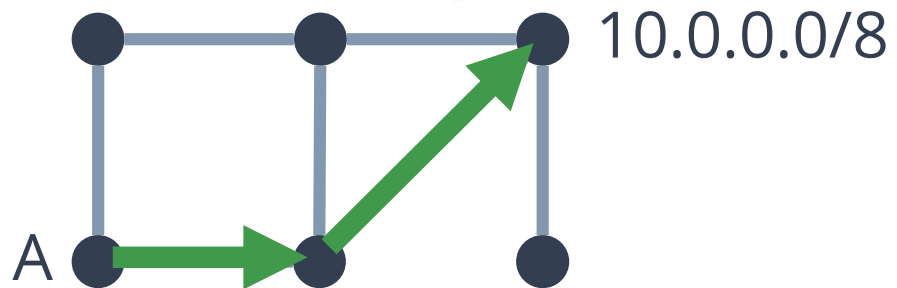
# Properties

Flow = ingress + destination



# Properties

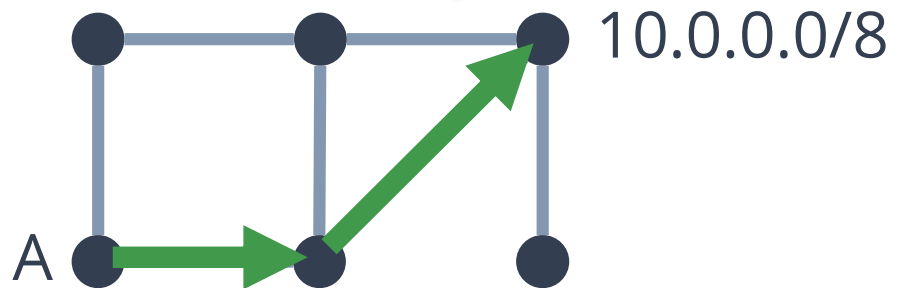
Flow = ingress + destination



Based on  
forwarding graph

# Properties

Flow = ingress + destination

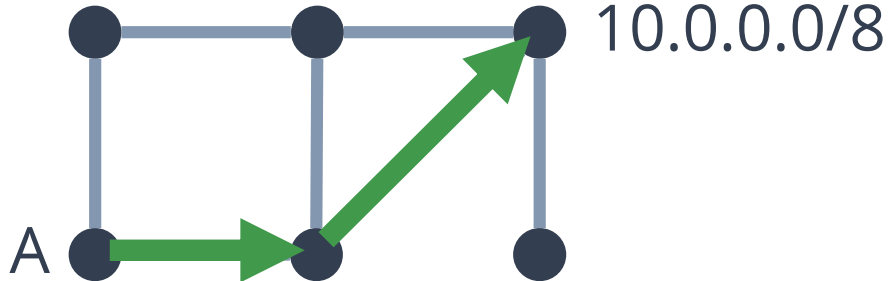


Based on  
forwarding graph

Reachability

# Properties

Flow = ingress + destination



Based on  
forwarding graph

*Single-flow* properties

Reachability

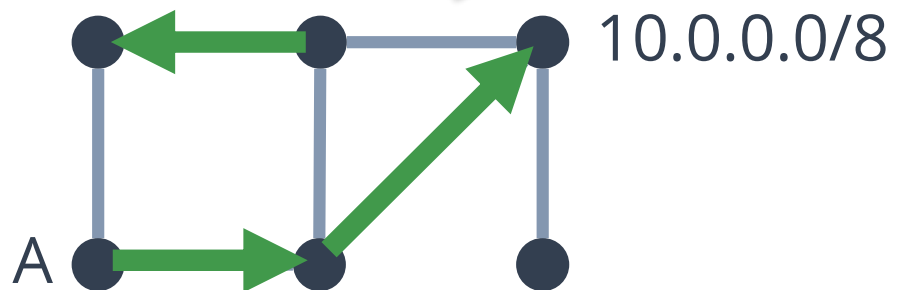
Path length

Waypointing

...

# Properties

Flow = ingress + destination



Based on  
forwarding graph

## *Single-flow* properties

Reachability

Path length

Waypointing

...

## *Multi-flow* properties

Isolation

...

# Properties

NetDice targets  
*few-flow* properties

## *Single-flow* properties

Reachability

Path length

Waypointing

...

## *Multi-flow* properties

Isolation

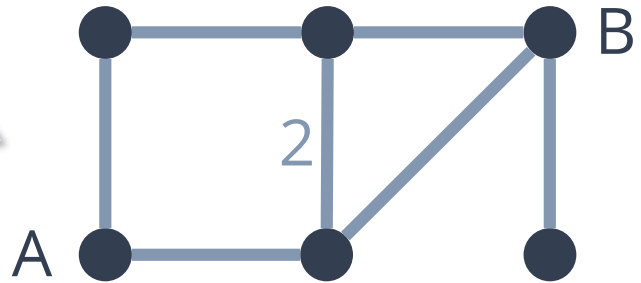
...

# Pruning Failures



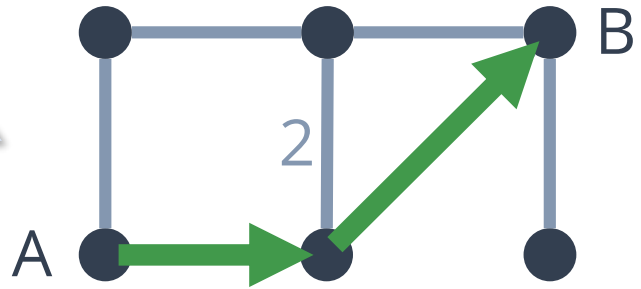
# Key Idea

shortest paths

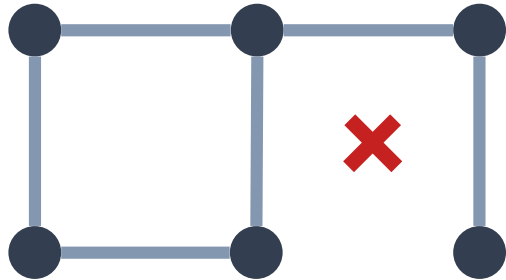
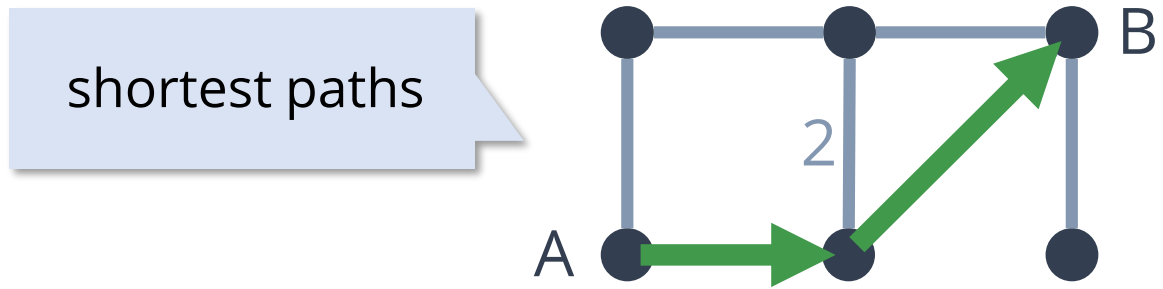


# Key Idea

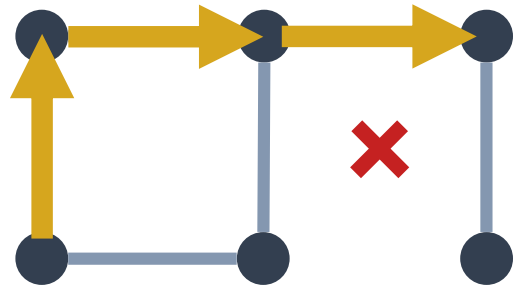
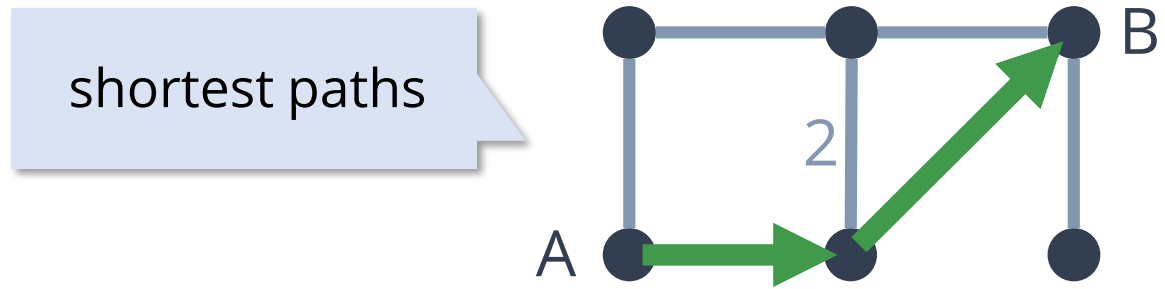
shortest paths



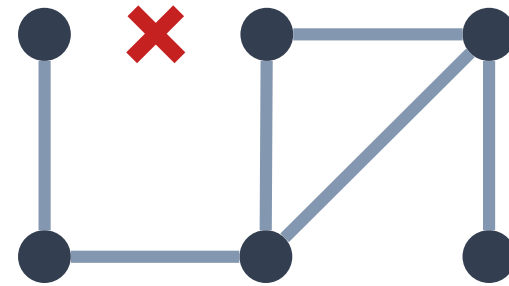
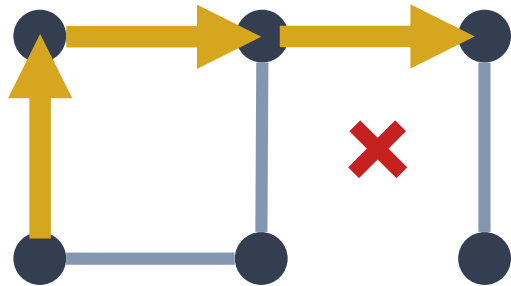
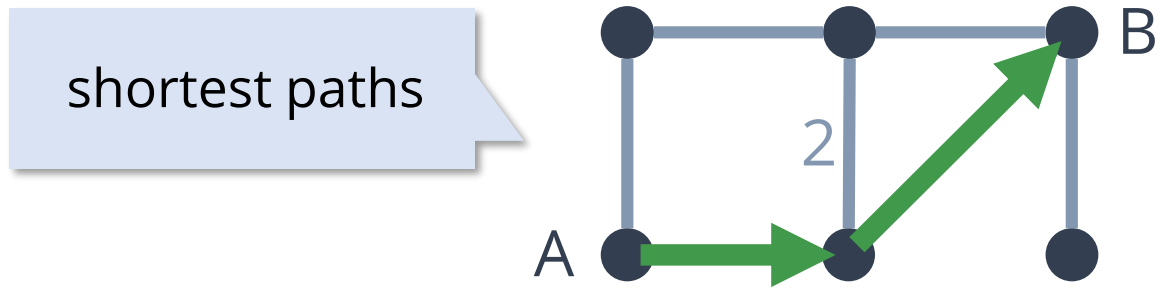
# Key Idea



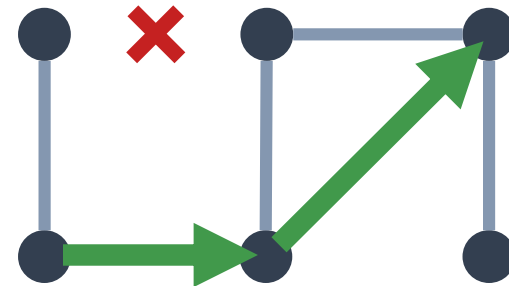
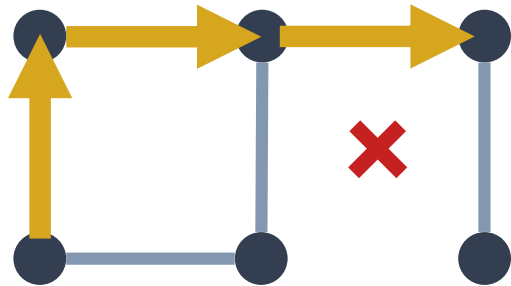
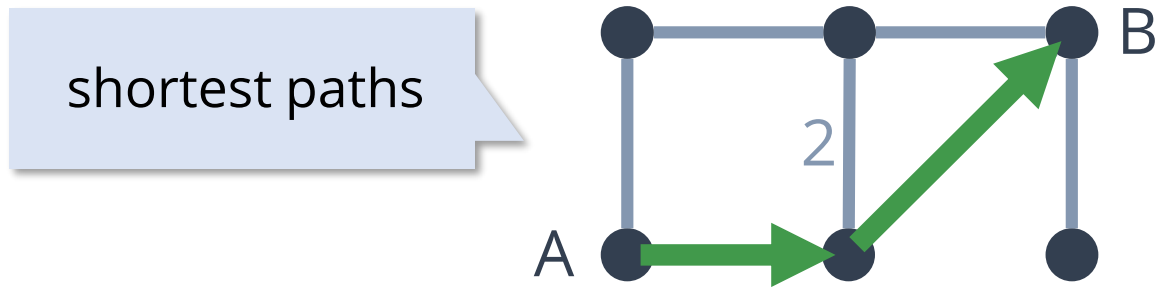
# Key Idea



# Key Idea

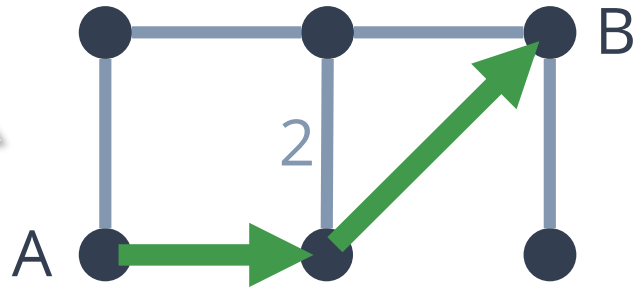


# Key Idea

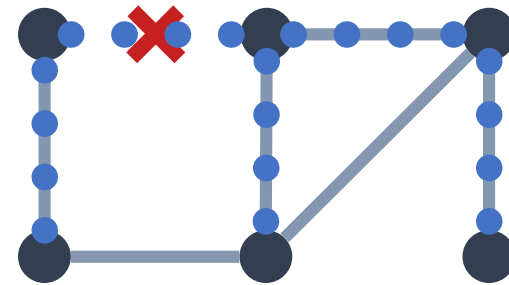
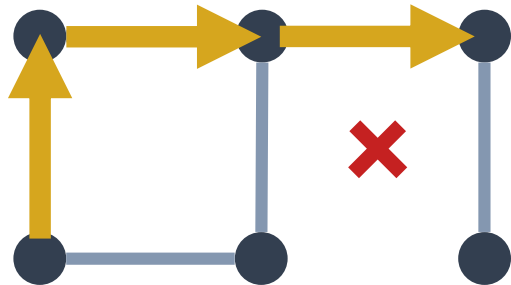


# Key Idea

shortest paths

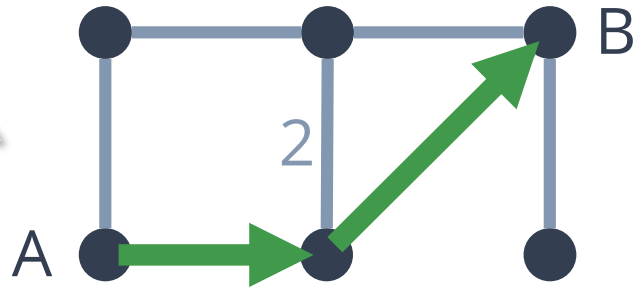


 cold edges

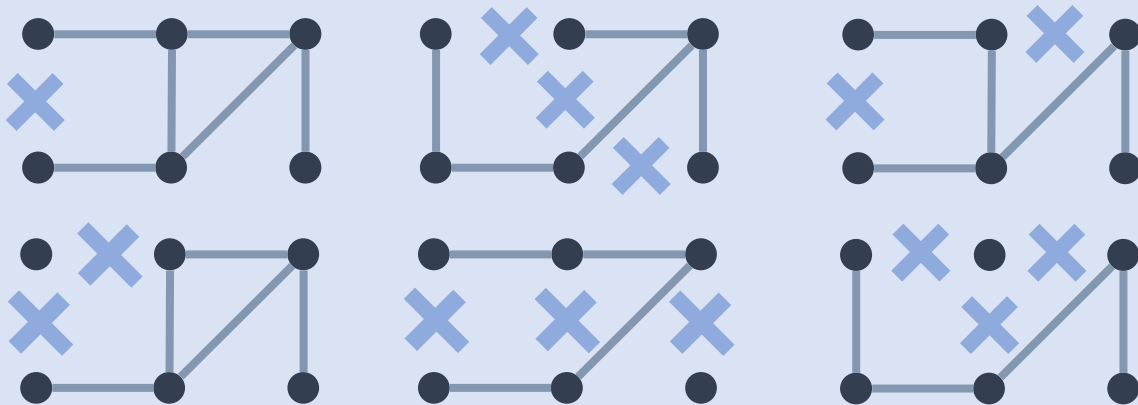


# Key Idea

shortest paths

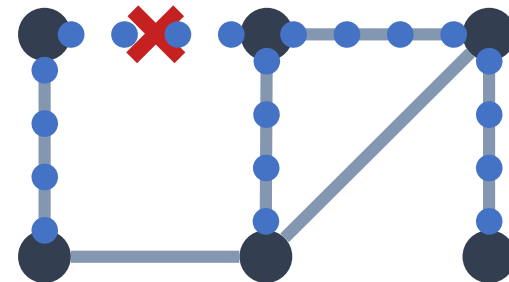


Scenarios with same forwarding graph (32 total):



...

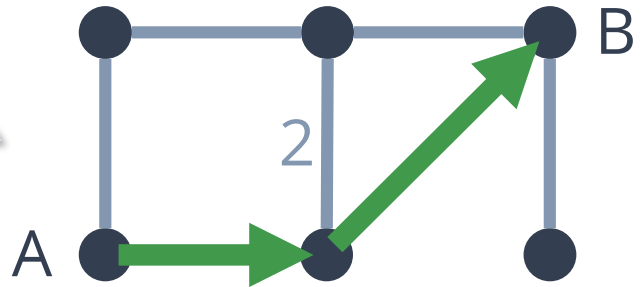
 cold edges





# Key Idea

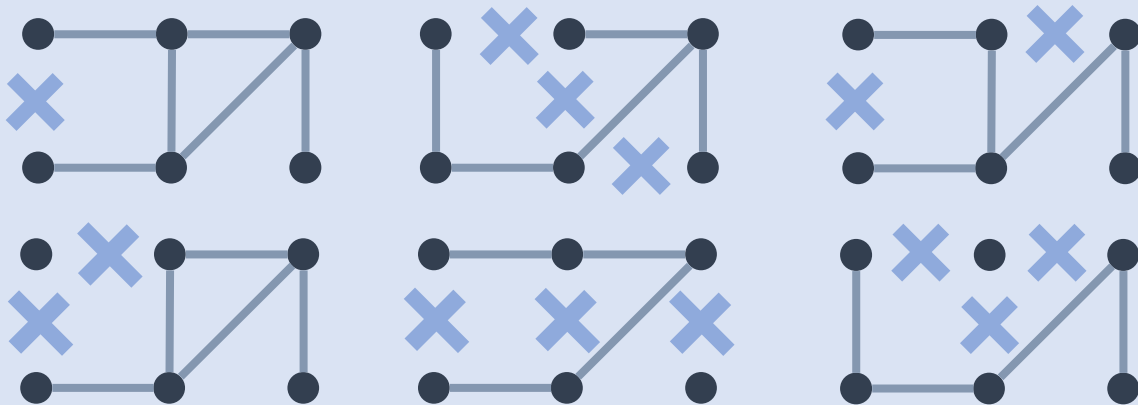
shortest paths



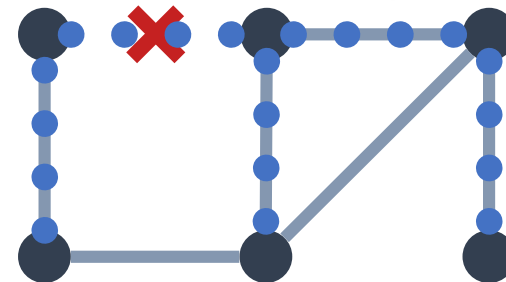
How to find these?

 cold edges

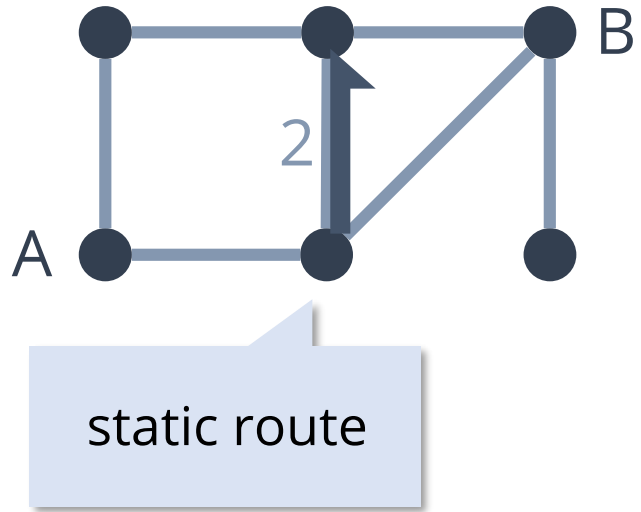
Scenarios with same forwarding graph (32 total):



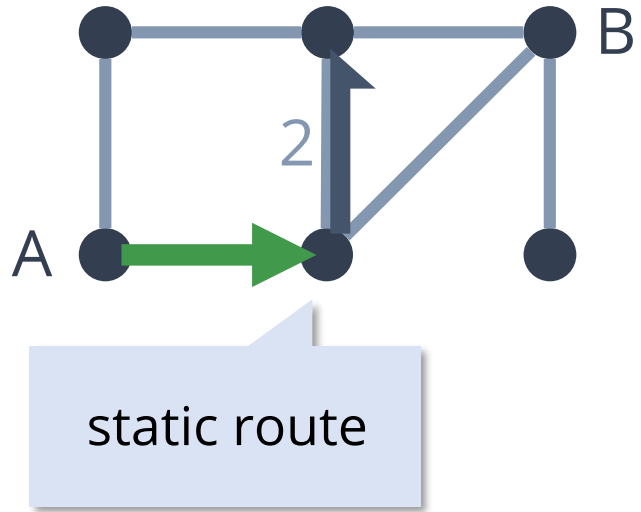
...



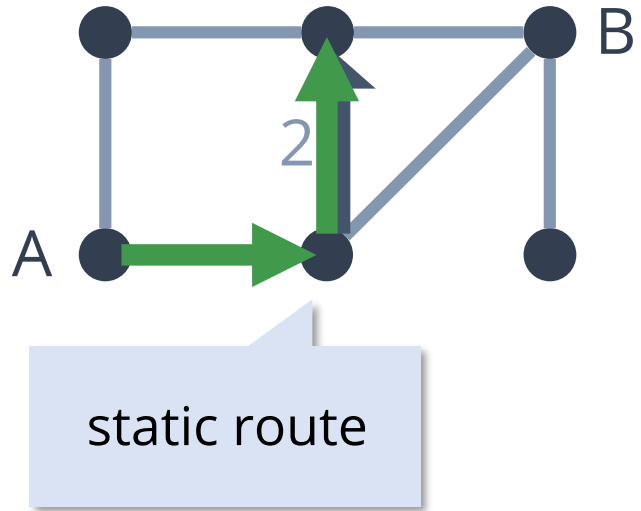
# ❄ for OSPF + static routes



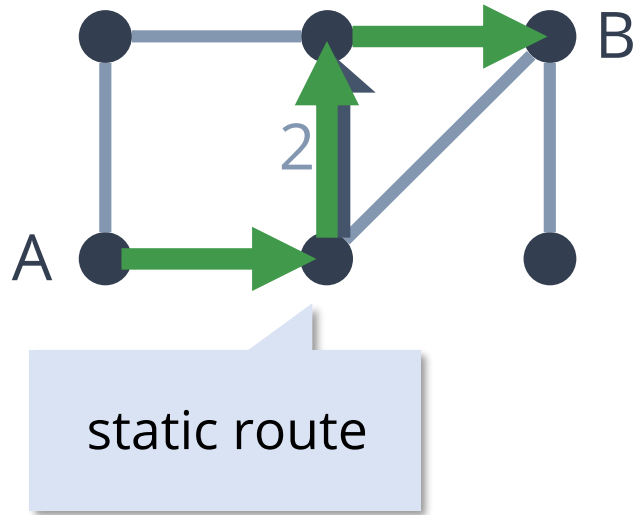
# ❄ for OSPF + static routes



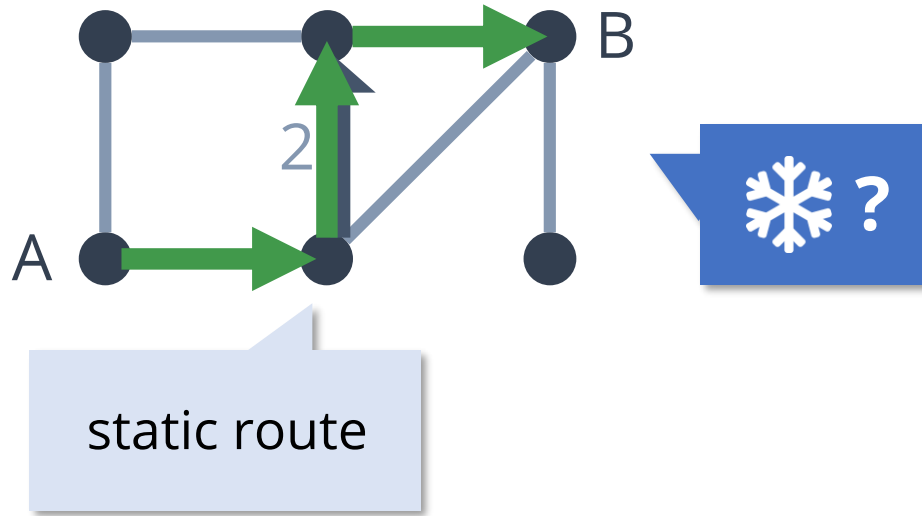
100



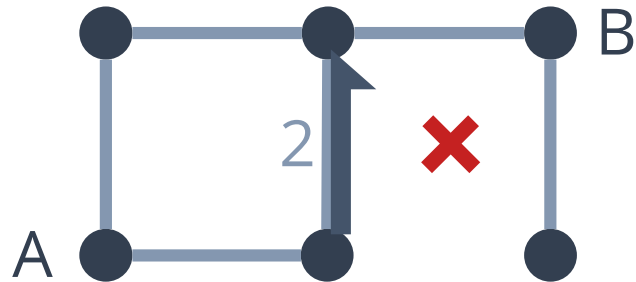
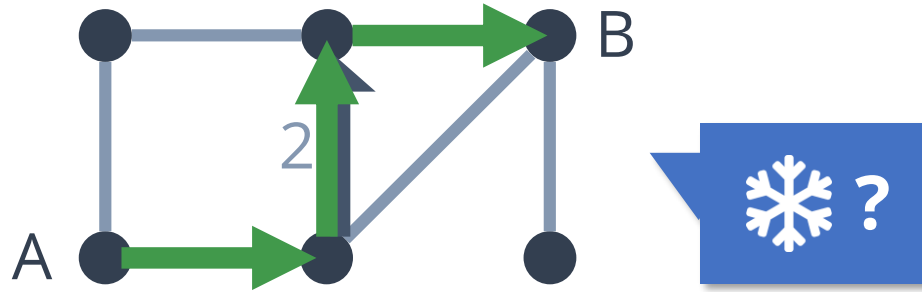
# ❄ for OSPF + static routes



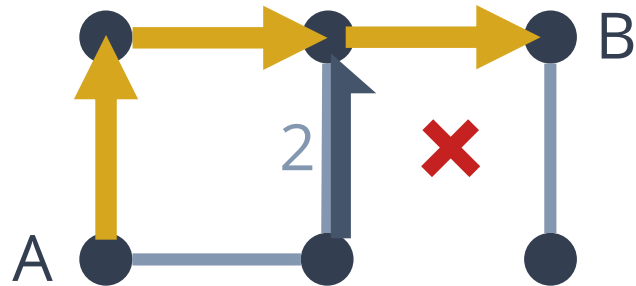
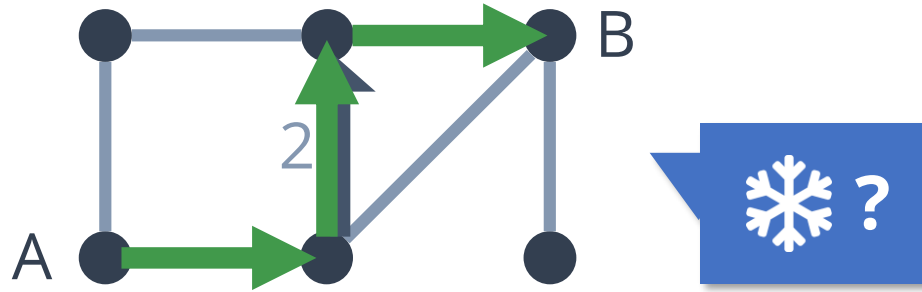
# ❄ for OSPF + static routes



# ❄️ for OSPF + static routes

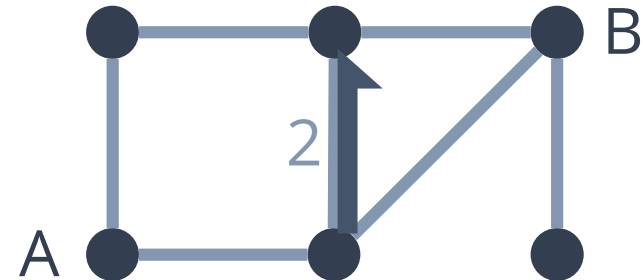
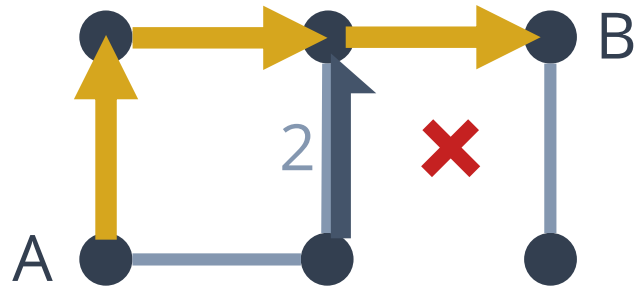
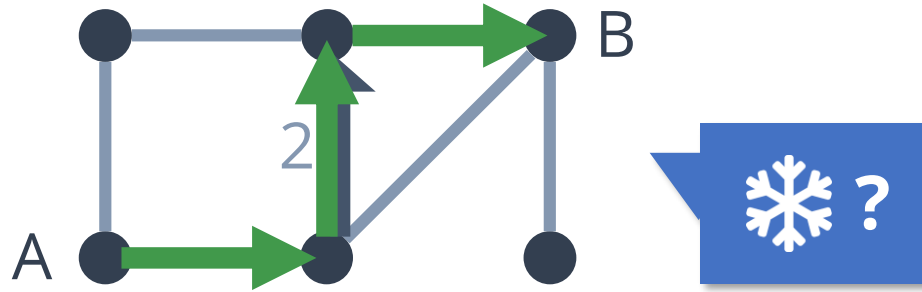


# ❄️ for OSPF + static routes

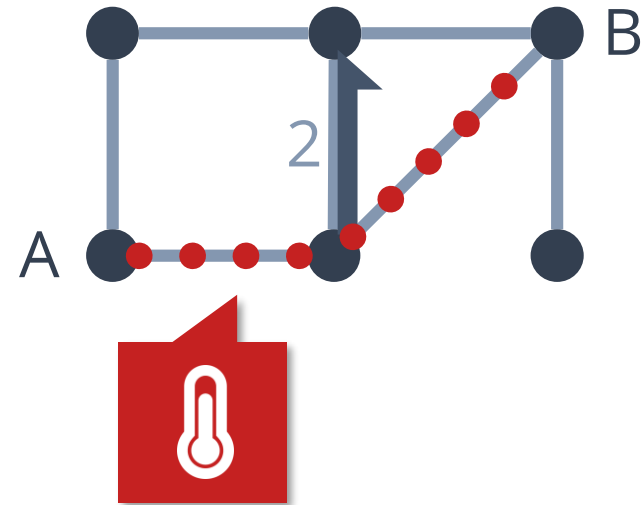
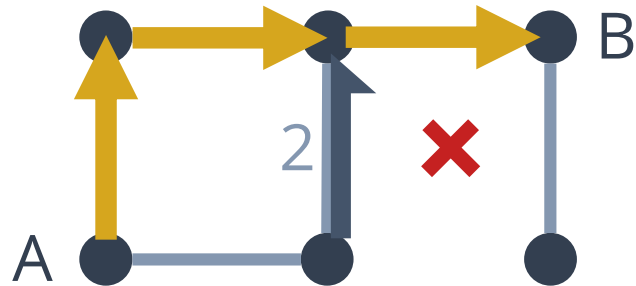
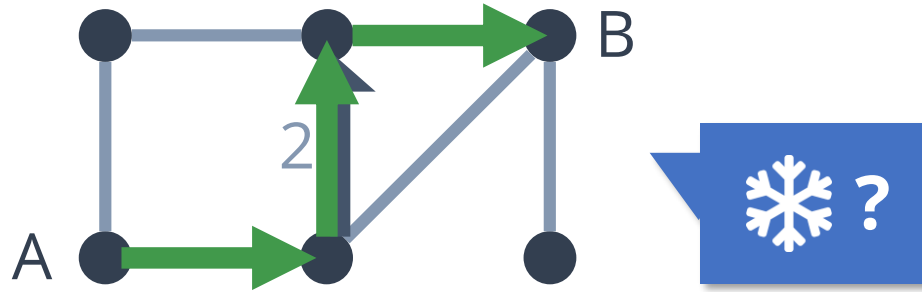




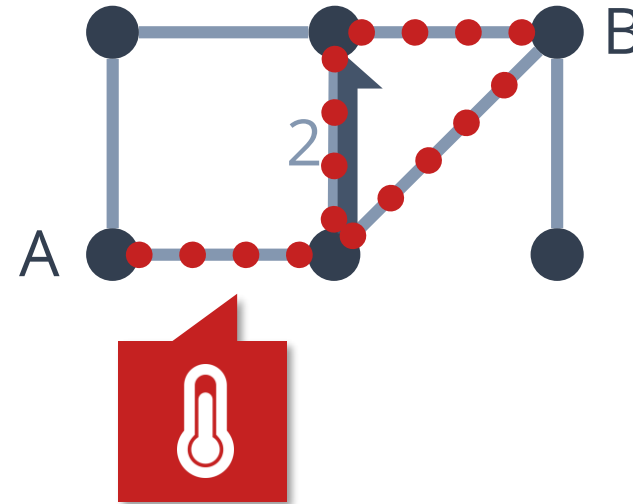
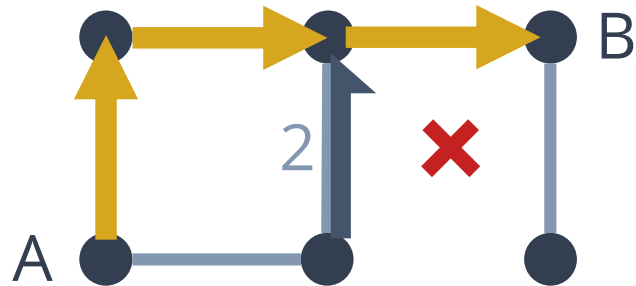
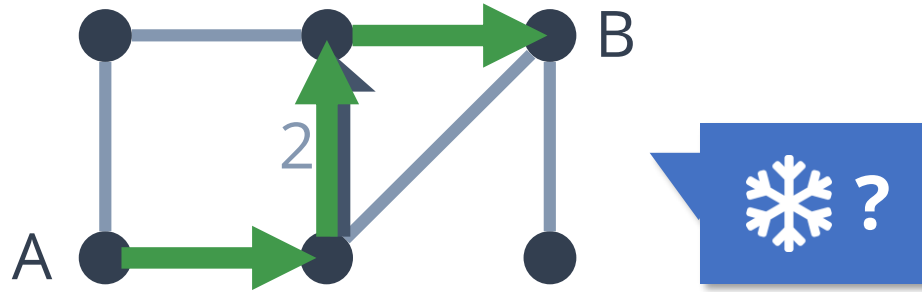
# ❄️ for OSPF + static routes



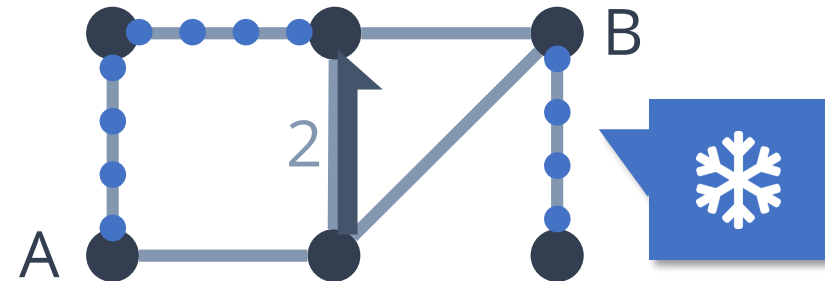
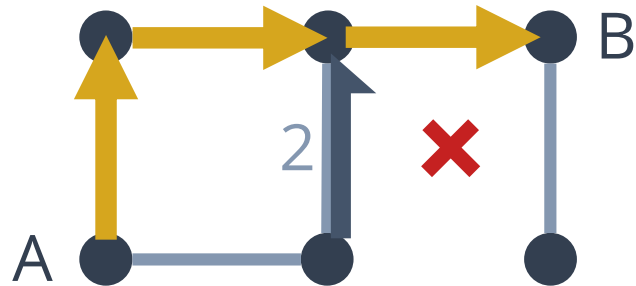
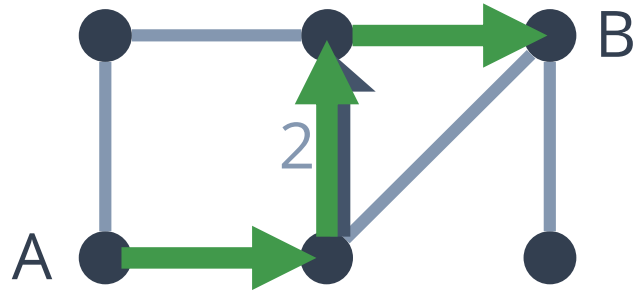
# ❄️ for OSPF + static routes



# ❄️ for OSPF + static routes



# ❄️ for OSPF + static routes



# ❄ for BGP

---

**Algorithm 3** Hot edges for BGP

---

```
1: procedure HOTBGP( $u, d, E_{\text{fwd}}, L$ )
2:    $X \leftarrow$  nodes in the same partition as  $u$  under  $L$ 
3:    $\text{BR}_L \leftarrow \text{TOP3}(\text{BR}, X)$  ▷ BGP pre-processing (§4.2)
4:    $\text{RR}_L \leftarrow \text{RR} \cap X$ 
5:    $\mathcal{H} \leftarrow \text{ALLSP}(\text{RR}_L, \text{BR}_L, L)$  ▷ all shortest paths (Alg. 2)
6:    $\mathcal{D} \leftarrow \{u\}$  ▷ decision points
7:      $\cup \{y \mid (x, y) \in \text{STATIC}_d \cap E_{\text{fwd}}\}$ 
8:      $\cup \{y \mid (x, y) \in E_{\text{fwd}} \wedge \text{NH}_d(x) \neq \text{NH}_d(y)\}$ 
9:   for each  $x \in \mathcal{D}$  do
10:     $\mathcal{H} \leftarrow \mathcal{H} \cup \text{SP}_L(x, \text{NH}_d(x))$  ▷ shortest path  $x \rightarrow \text{NH}_d(x)$ 
11:     $\mathcal{H} \leftarrow \mathcal{H} \cup (\text{STATIC}_d \cap E_{\text{fwd}})$  ▷ traversed static routes
12:   if  $\text{RR}_L = \emptyset$  then
13:     $\mathcal{H} \leftarrow \mathcal{H} \cup \text{ALLSP}(\{u\}, \text{BR}_L)$  ▷ ensure connectivity
14:   return  $\mathcal{H}$ 
```

---

see paper

# ❄️ for BGP

---

**Algorithm 3** Hot edges for BGP

---

```
1: procedure HOTBGP( $u, d, E_{\text{fwd}}, L$ )
2:    $X \leftarrow$  nodes in the same partition as  $u$  under  $L$ 
3:    $\text{BR}_L \leftarrow \text{TOP3}(\text{BR}, X)$  ▷ BGP pre-processing (§4.2)
4:    $\text{RR}_L \leftarrow \text{RR} \cap X$ 
5:    $\mathcal{H} \leftarrow \text{ALLSP}(\text{RR}_L, \text{BR}_L, L)$  ▷ all shortest paths (Alg. 2)
6:    $\mathcal{D} \leftarrow \{u\}$  ▷ decision points
7:      $\cup \{y \mid (x, y) \in \text{STATIC}_d \cap E_{\text{fwd}}\}$ 
8:      $\cup \{y \mid (x, y) \in E_{\text{fwd}} \wedge \text{NH}_d(x) \neq \text{NH}_d(y)\}$ 
9:   for each  $x \in \mathcal{D}$  do
10:     $\mathcal{H} \leftarrow \mathcal{H} \cup \text{SP}_L(x, \text{NH}_d(x))$  ▷ shortest path  $x \rightarrow \text{NH}_d(x)$ 
11:     $\mathcal{H} \leftarrow \mathcal{H} \cup (\text{STATIC}_d \cap E_{\text{fwd}})$  ▷ traversed static routes
12:   if  $\text{RR}_L = \emptyset$  then
13:     $\mathcal{H} \leftarrow \mathcal{H} \cup \text{ALLSP}(\{u\}, \text{BR}_L)$  ▷ ensure connectivity
14:   return  $\mathcal{H}$ 
```

---

see paper

network partitions

route reflection

dependence on  
IGP costs

# ❄️ for BGP

---

**Algorithm 3** Hot edges for BGP

---

```
1: procedure HOTBGP( $u, d, E_{\text{fwd}}, L$ )
2:    $X \leftarrow$  nodes in the same partition as  $u$  under  $L$ 
3:    $\text{BR}_L \leftarrow \text{TOP3}(\text{BR}, X)$  ▷ BGP pre-processing (§4.2)
4:    $\text{RR}_L \leftarrow \text{RR} \cap X$ 
5:    $\mathcal{H} \leftarrow \text{ALLSP}(\text{RR}_L, \text{BR}_L, L)$  ▷ all shortest paths (Alg. 2)
6:    $\mathcal{D} \leftarrow \{u\}$  ▷ decision points
7:      $\cup \{y \mid (x, y) \in \text{STATIC}_d \cap E_{\text{fwd}}\}$ 
8:      $\cup \{y \mid (x, y) \in E_{\text{fwd}} \wedge \text{NH}_d(x) \neq \text{NH}_d(y)\}$ 
9:   for each  $x \in \mathcal{D}$  do
10:     $\mathcal{H} \leftarrow \mathcal{H} \cup \text{SP}_L(x, \text{NH}_d(x))$  ▷ shortest path  $x \rightarrow \text{NH}_d(x)$ 
11:     $\mathcal{H} \leftarrow \mathcal{H} \cup (\text{STATIC}_d \cap E_{\text{fwd}})$  ▷ traversed static
12:    if  $\text{RR}_L = \emptyset$  then
13:       $\mathcal{H} \leftarrow \mathcal{H} \cup \text{ALLSP}(\{u\}, \text{BR}_L)$  ▷ ensure connectivity
14:  return  $\mathcal{H}$ 
```

---

see paper

network partitions

route reflection

dependence on  
IGP costs

with correctness proof



# Failure Model



# Failure Model

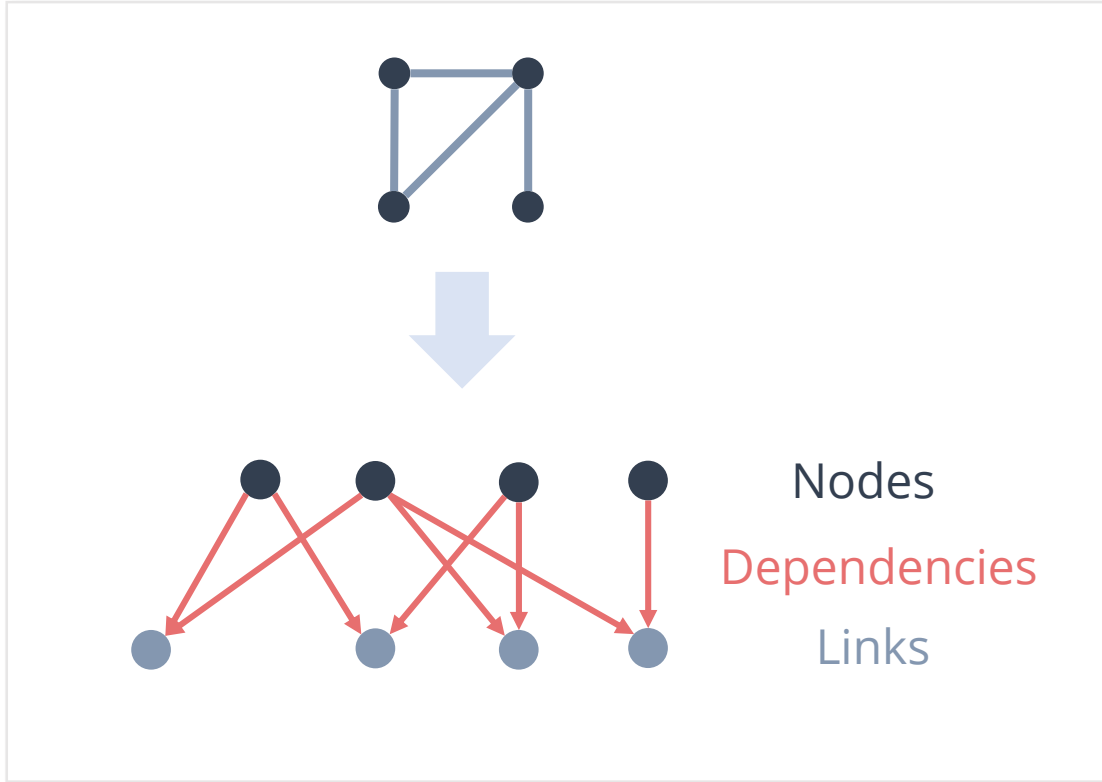
Bayesian network

# Failure Model

Bayesian network

Allows arbitrary dependencies

# Failure Model

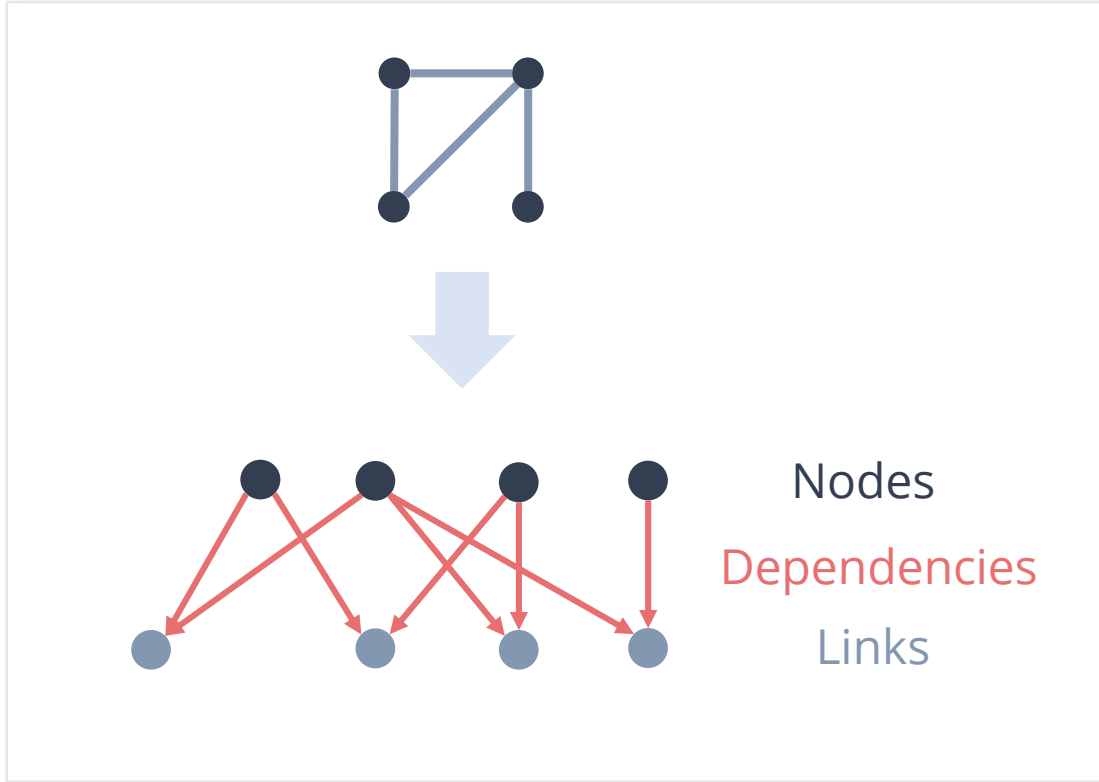


Bayesian network

Allows arbitrary dependencies

Link and node failures

# Failure Model



Bayesian network

Allows arbitrary dependencies

Inference using *Variable Elimination*

Link and node failures

# Exploring Failures

# Computing $P(\text{🐻✅})$

$$P(\text{🐻✅}) = \sum_s P(\text{🐻✅} | s) \cdot P(s)$$

failure scenario

# Computing $P(\text{🐼}✓)$

$$P(\text{🐼}✓) = \sum_s P(\text{🐼}✓ | s) \cdot P(s)$$

0 or 1

failure scenario

# Computing $P(\text{pass})$

$$P(\text{pass}) = \sum_s P(\text{pass} | s) \cdot P(s)$$

0 or 1

from failure model

failure scenario



# Computing $P(\text{pass})$

$$P(\text{pass}) = \sum_s P(\text{pass} | s) \cdot P(s)$$

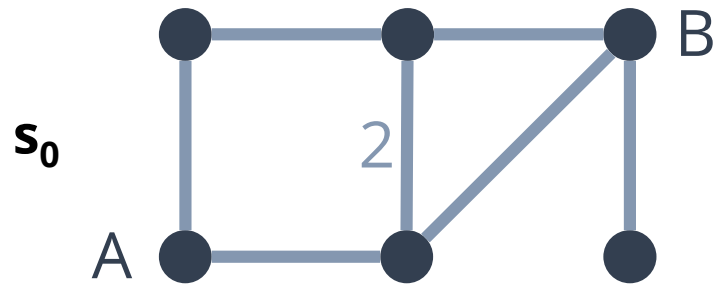
0 or 1

from failure model

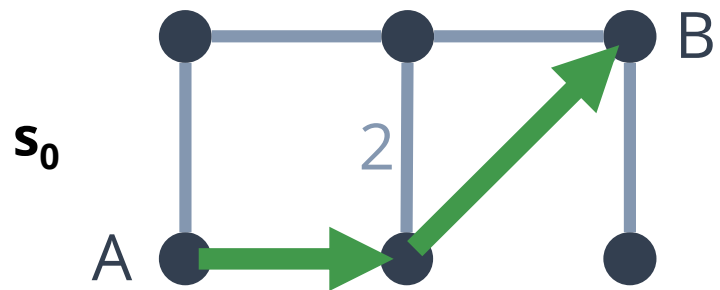
failure scenario

$$= \sum_{s \text{ s.t. } \text{pass}} P(s)$$

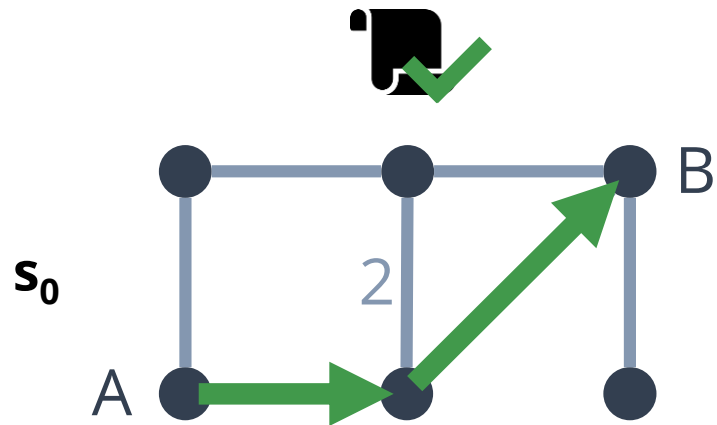
# Failure Exploration



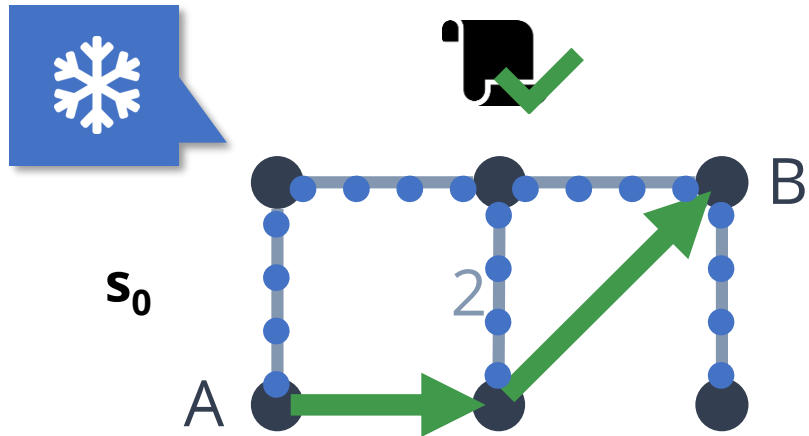
# Failure Exploration



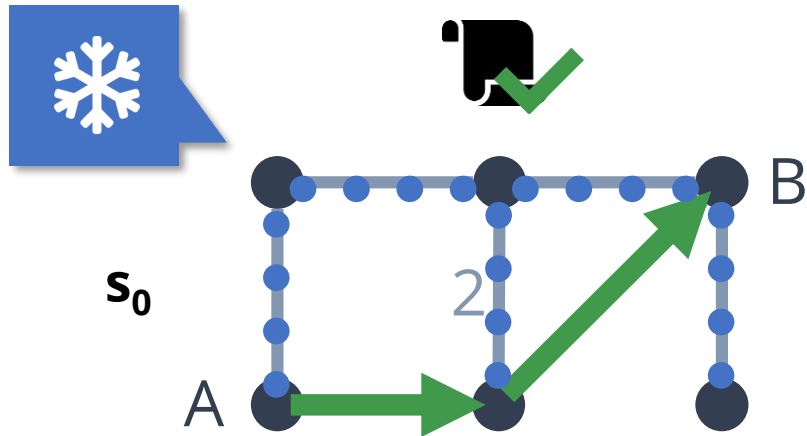
# Failure Exploration



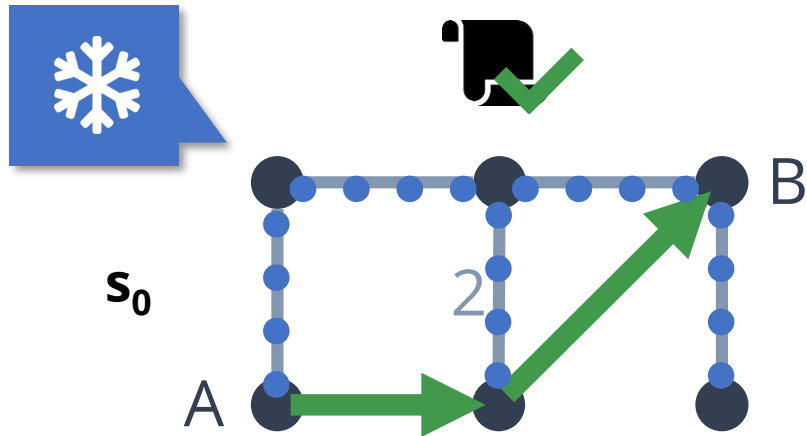
# Failure Exploration



# Failure Exploration



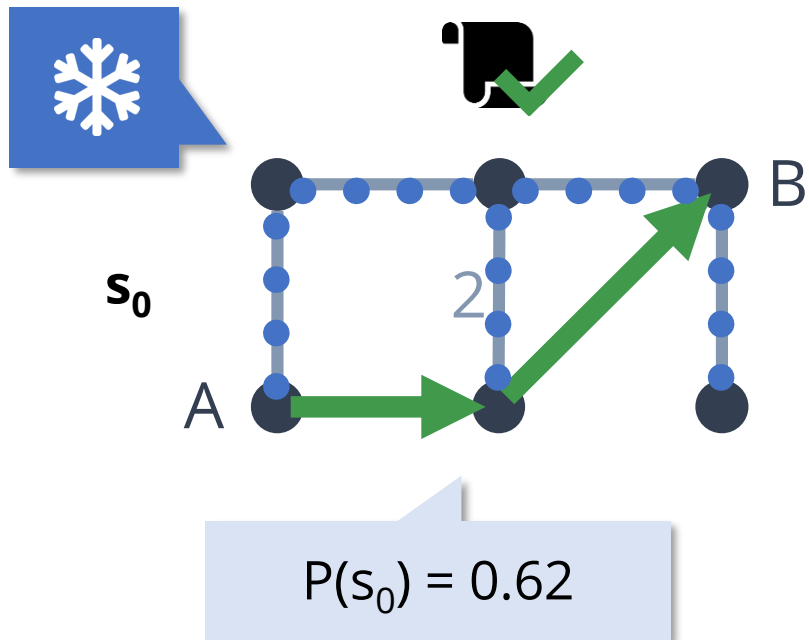
# Failure Exploration



Compute probability mass of

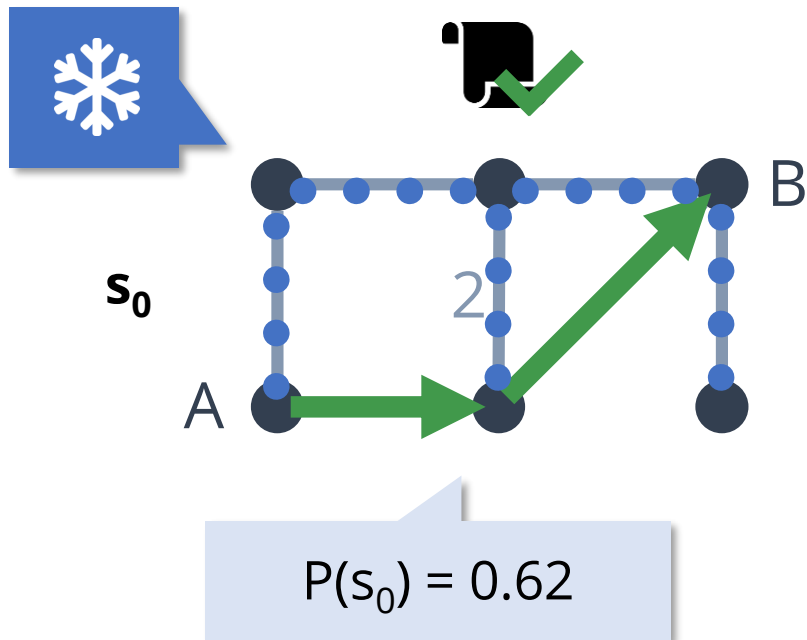
$$s_0 = \{ \text{[grid graph]}, \text{[grid graph with blue 'x' marks]}, \text{[grid graph with blue 'x' marks]}, \dots \}$$

# Failure Exploration

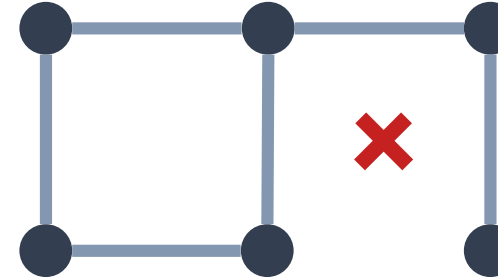




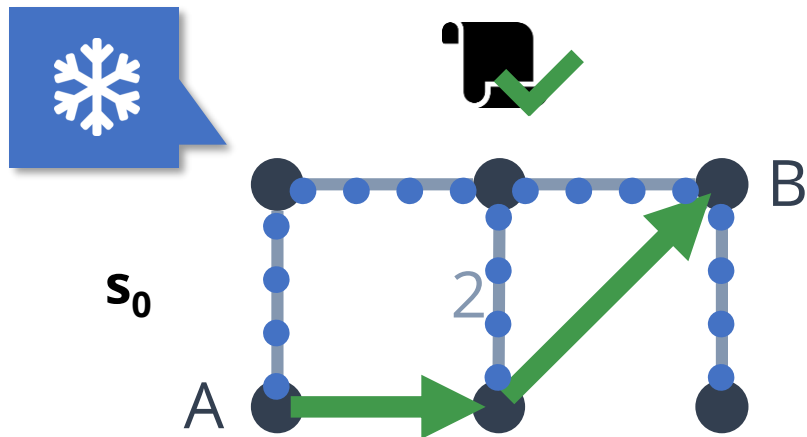
# Failure Exploration



$s_1$



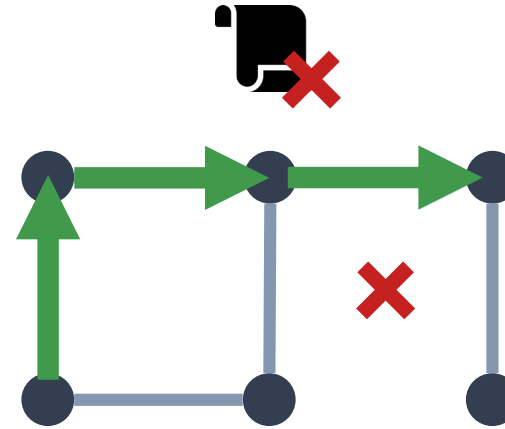
# Failure Exploration



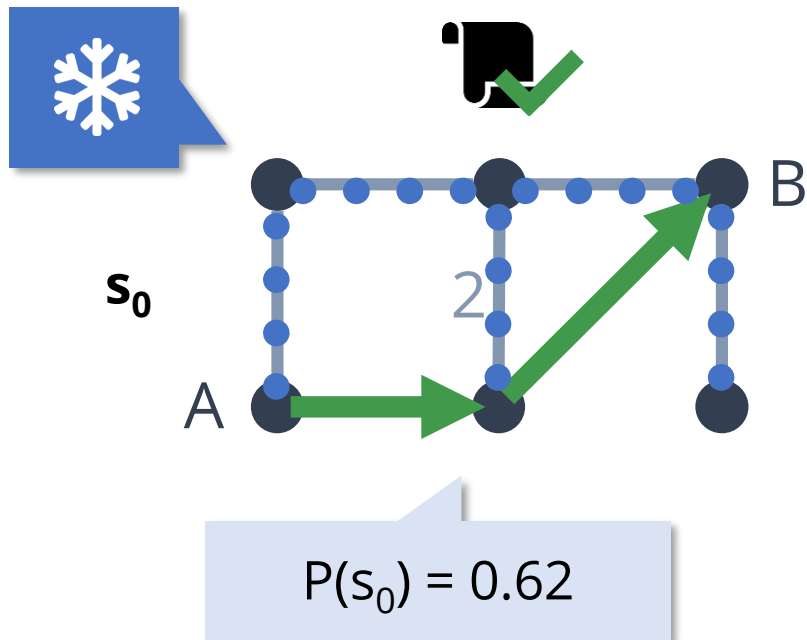
$$P(s_0) = 0.62$$



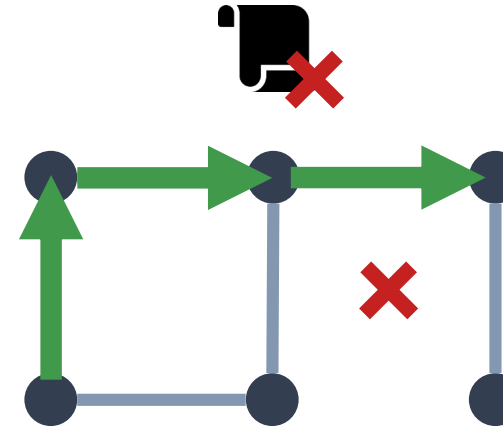
$s_1$



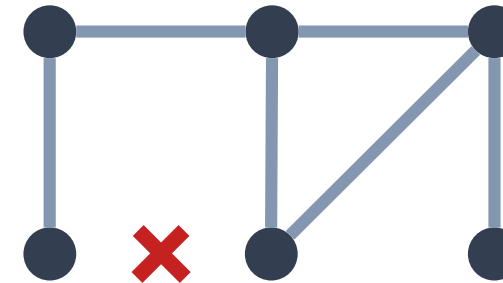
# Failure Exploration



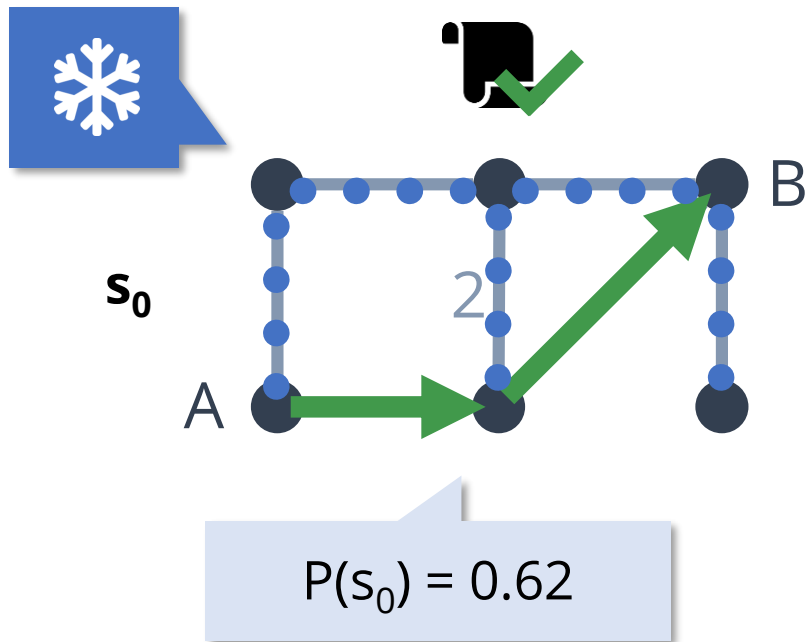
$s_1$



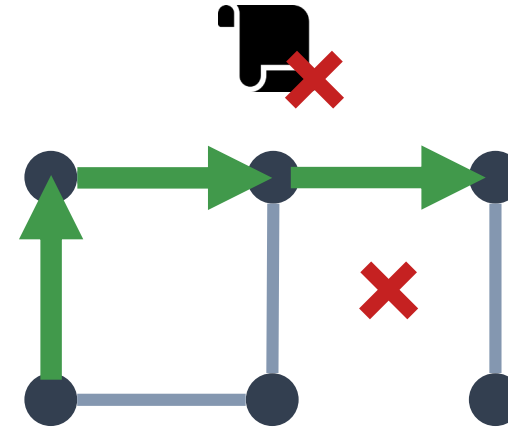
$s_2$



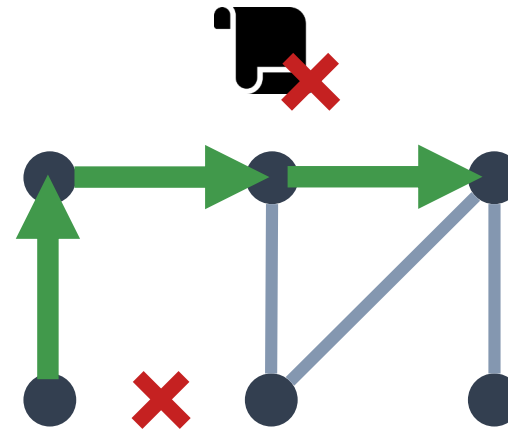
# Failure Exploration



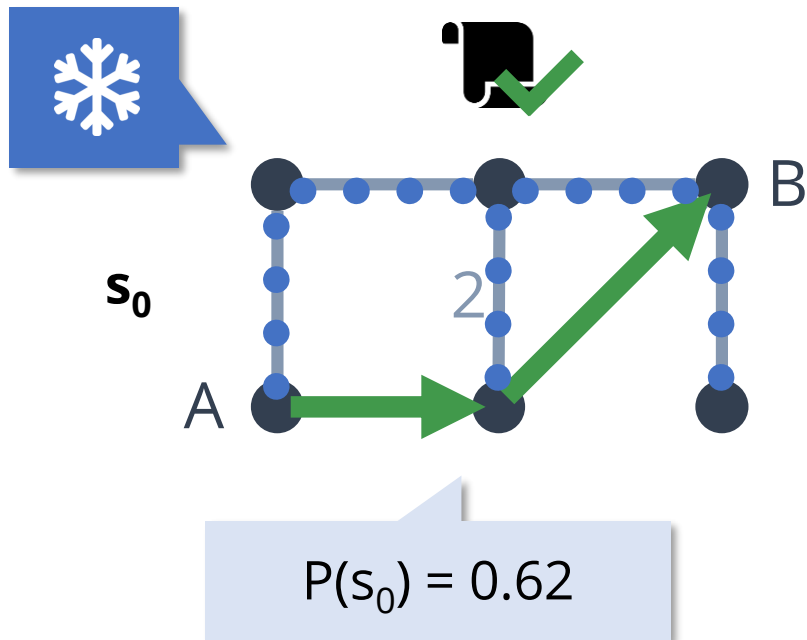
$s_1$



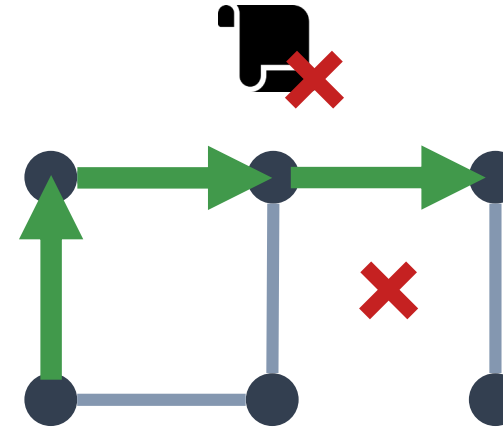
$s_2$



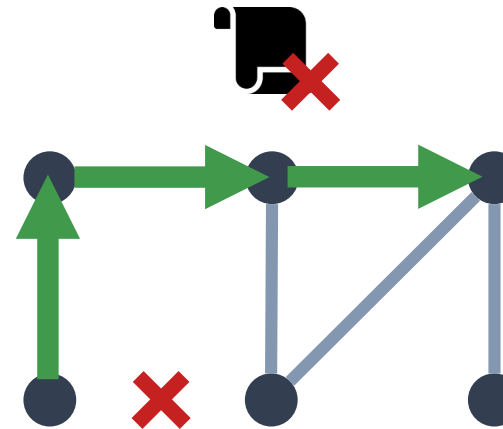
# Failure Exploration



$s_1$



$s_2$



...

...

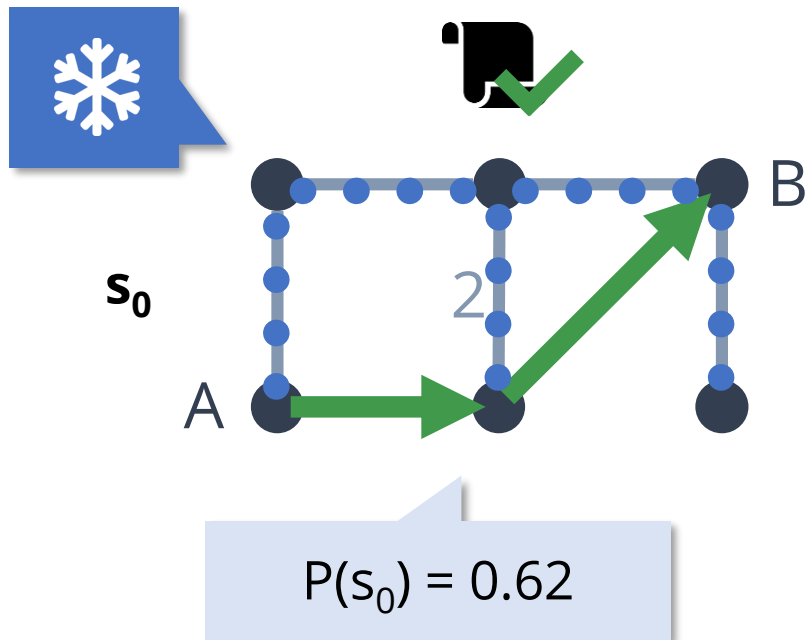
...

...

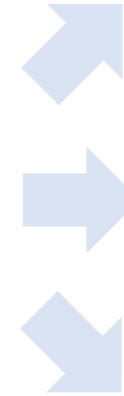
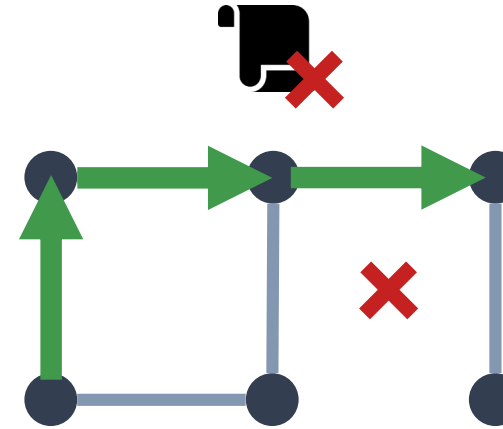
...

...

# Failure Exploration



$s_1$

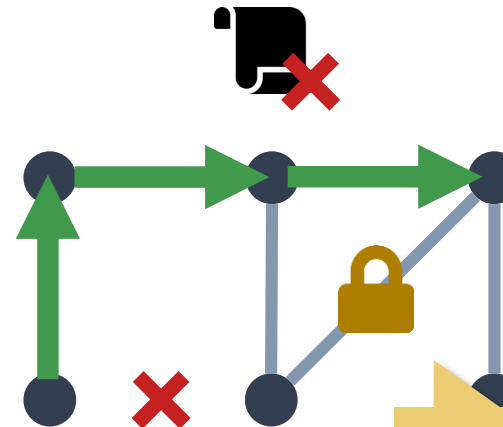


...

...

...

$s_2$



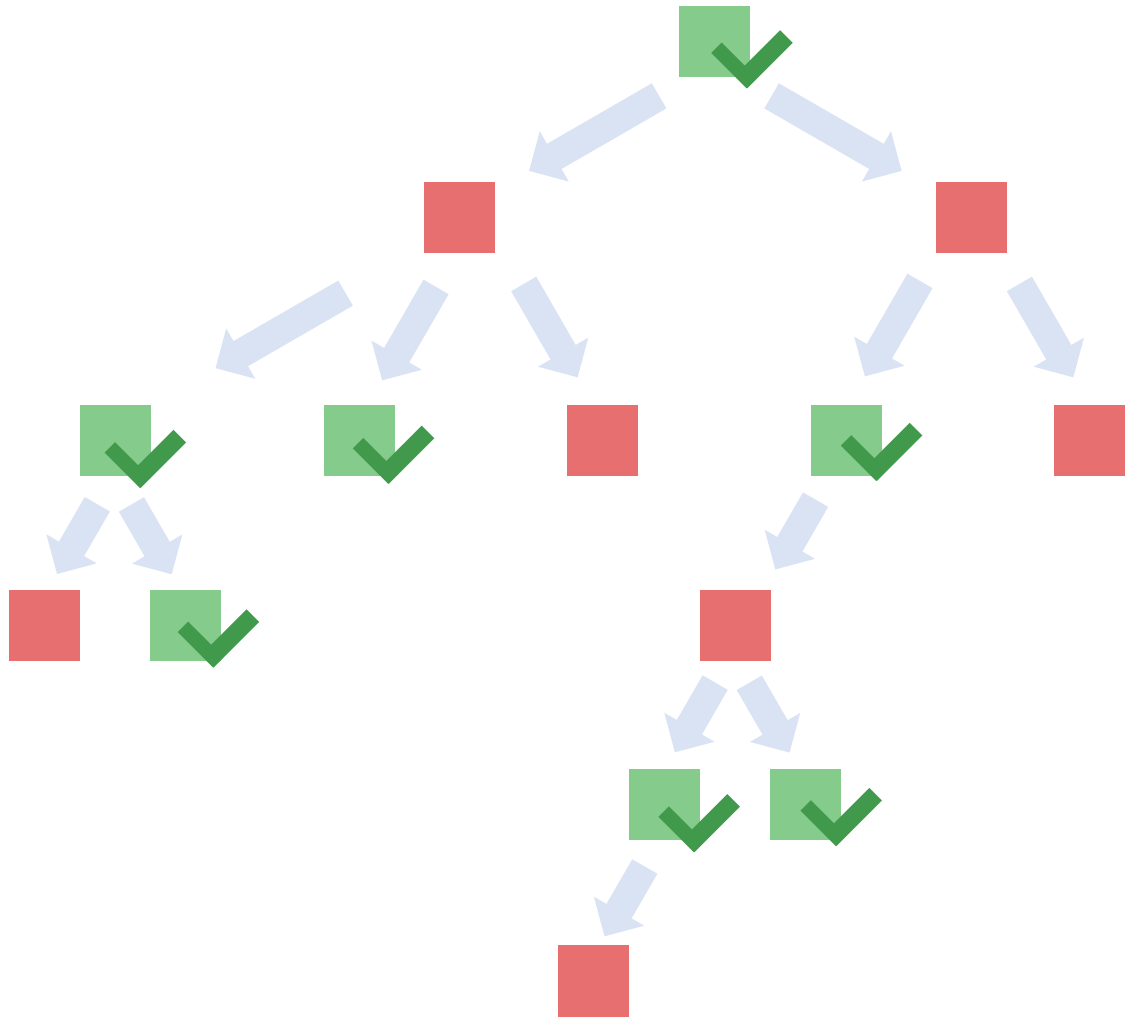
...

...

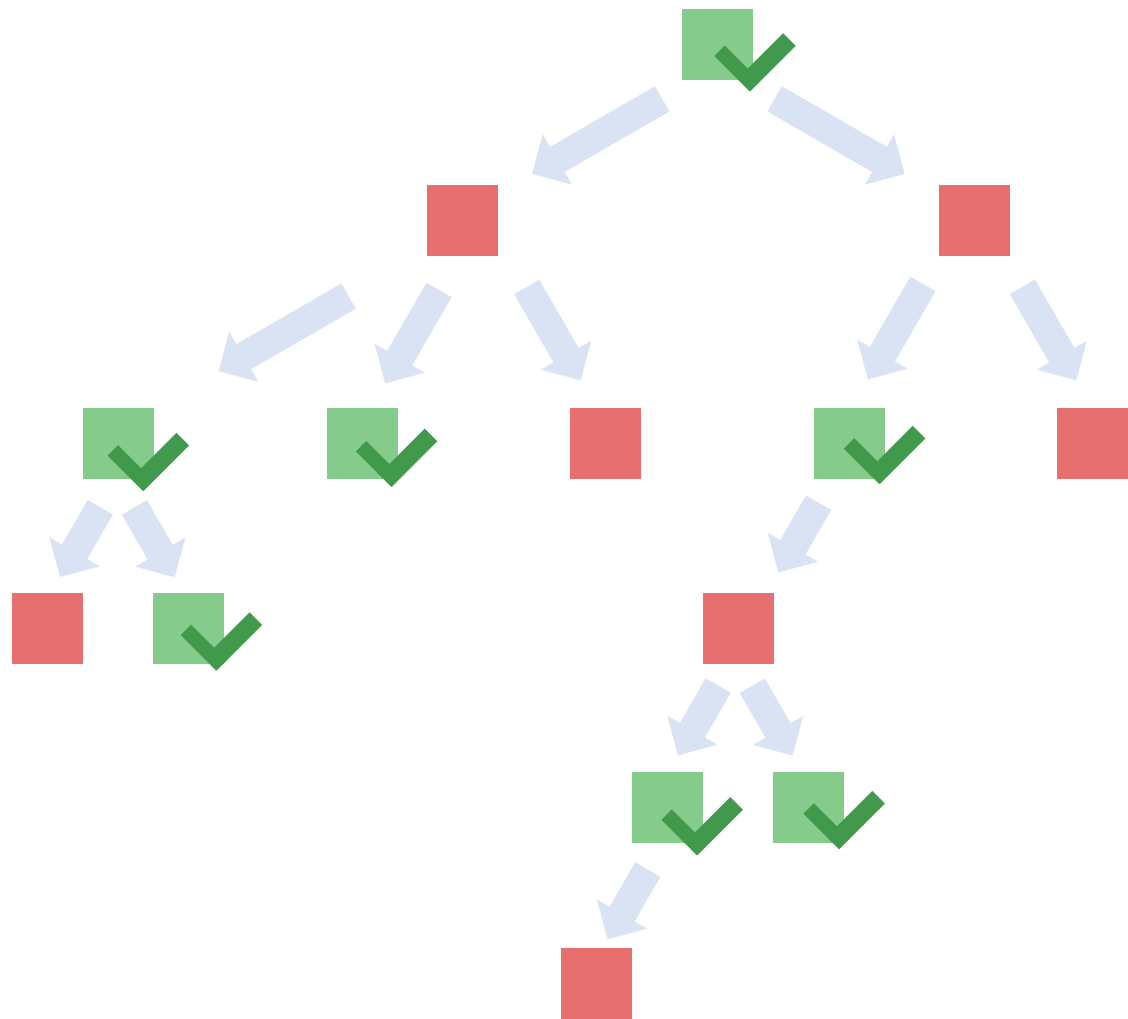
...

Don't explore  
duplicates

# Failure Exploration



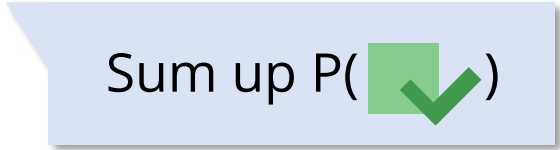
# Failure Exploration



Sum up  $P(\text{Green } \checkmark)$

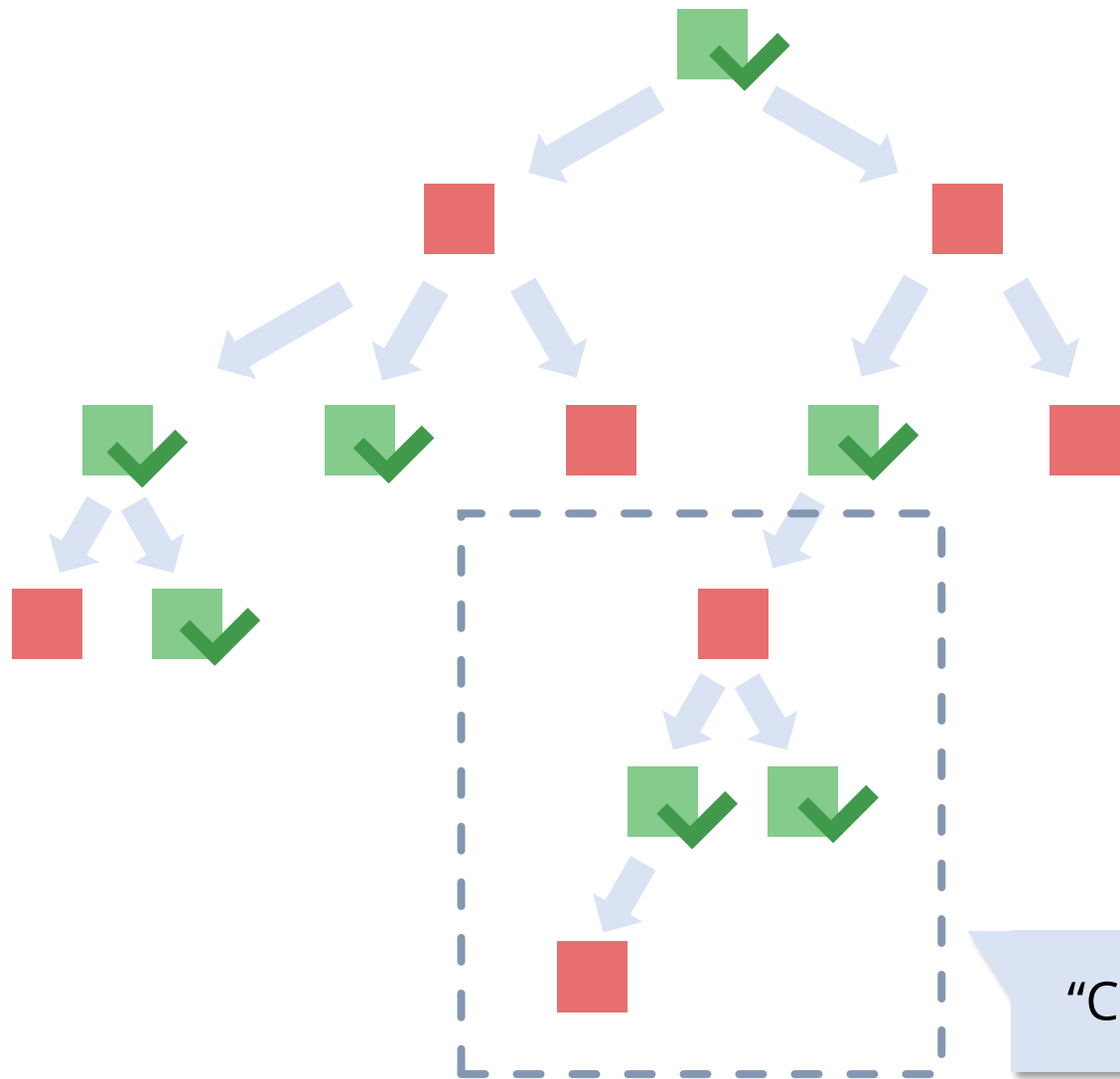


100



“Cut off” unlikely scenarios

# Failure Exploration



Sum up  $P(\text{Green } \checkmark)$

Efficiency depends on # ❄️

Very efficient in practice

"Cut off" unlikely scenarios

# Evaluation

# Implementation



Reachability

Path length

Egress

Waypointing

Isolation

Load balancing

Congestion

...

# Evaluation

**90 topologies** from Topology Zoo [\[Knight et al.\]](#) and mrinfo probing [\[Mérindol et al.\]](#)

# Evaluation

**90 topologies** from Topology Zoo [\[Knight et al.\]](#) and mrinfo probing [\[Mérindol et al.\]](#)

50 – 2320 links

$p_{\text{link}} = 0.001, p_{\text{node}} = 0.0001$

# Evaluation

**90 topologies** from Topology Zoo [\[Knight et al.\]](#) and mrinfo probing [\[Mérindol et al.\]](#)

50 – 2320 links

$p_{\text{link}} = 0.001, p_{\text{node}} = 0.0001$

Synthetic BGP configurations

2 route reflectors, 10 border routers

# Evaluation

**90 topologies** from Topology Zoo [\[Knight et al.\]](#) and mrinfo probing [\[Mérindol et al.\]](#)

50 – 2320 links

$p_{\text{link}} = 0.001, p_{\text{node}} = 0.0001$

Synthetic BGP configurations

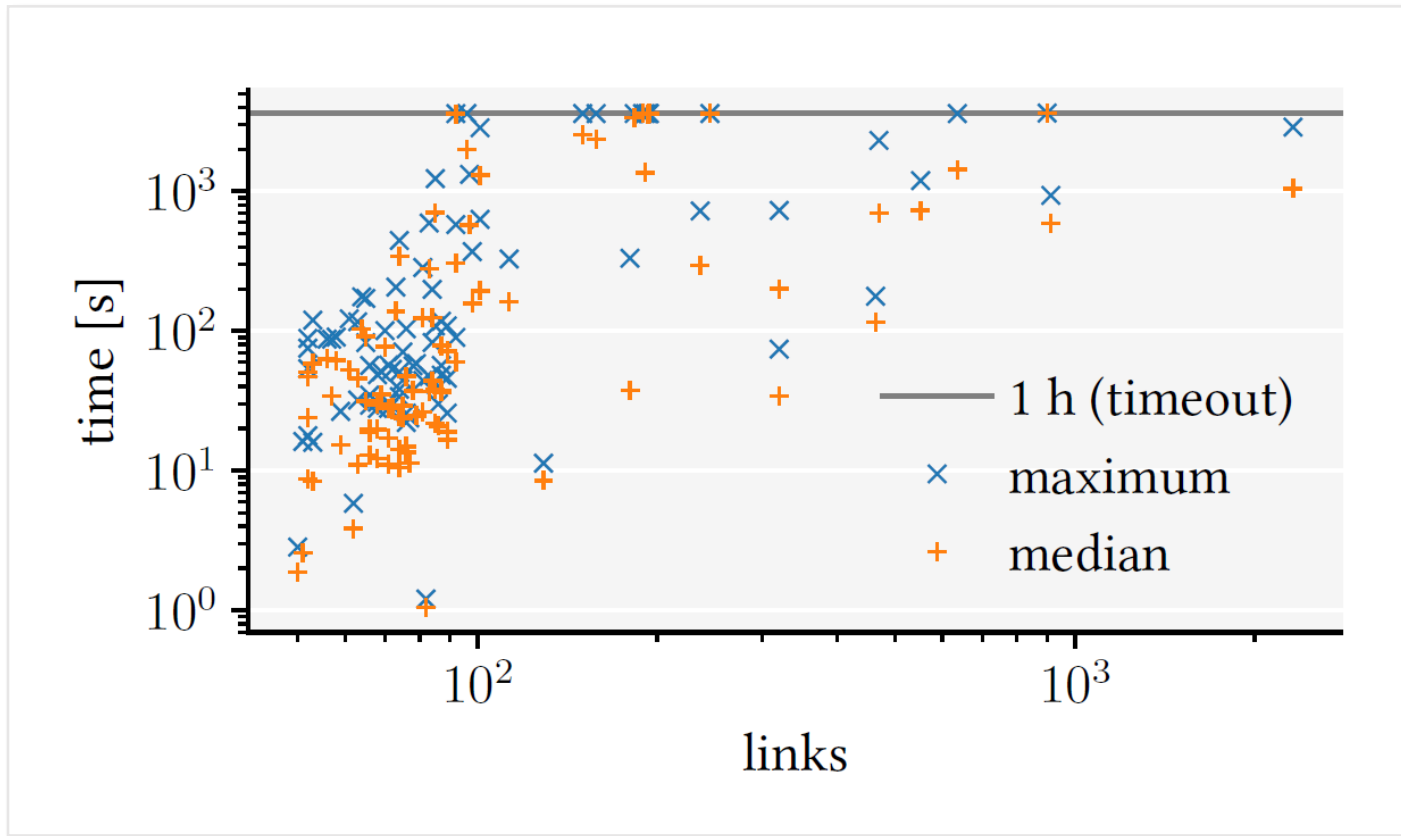
2 route reflectors, 10 border routers

Real ISP configuration

See paper

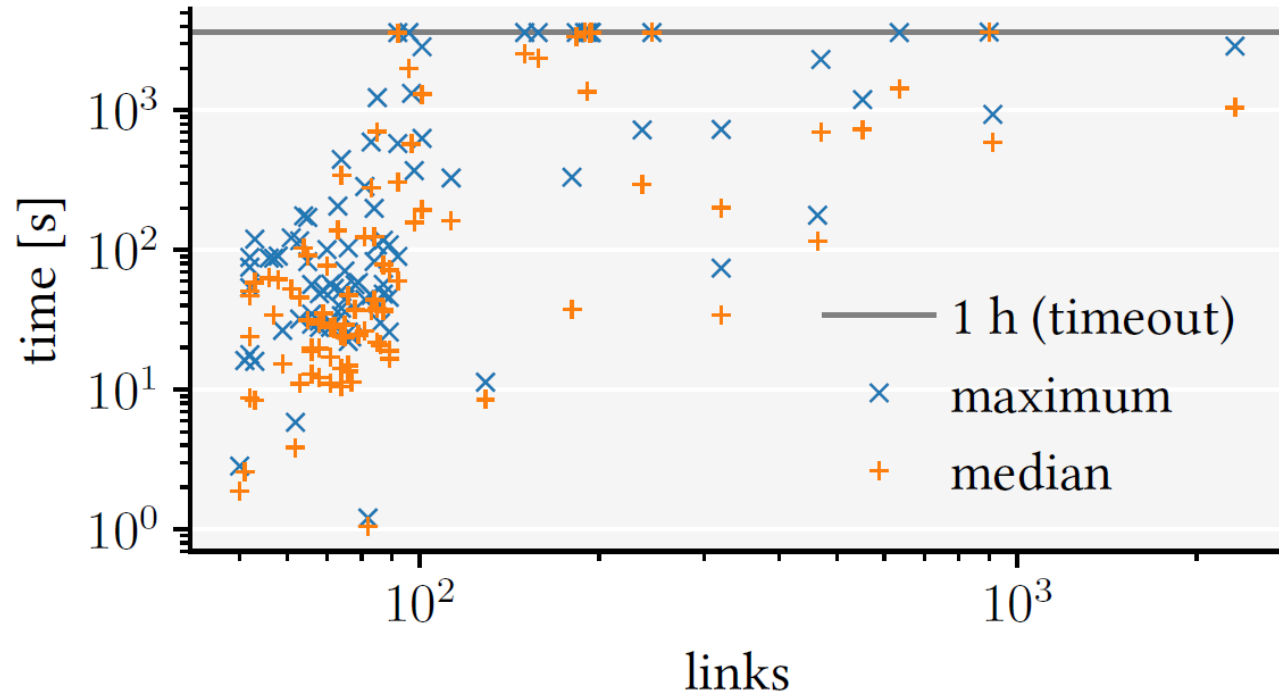


# Single-flow Properties



Waypoint, *four 9s* precision

# Single-flow Properties

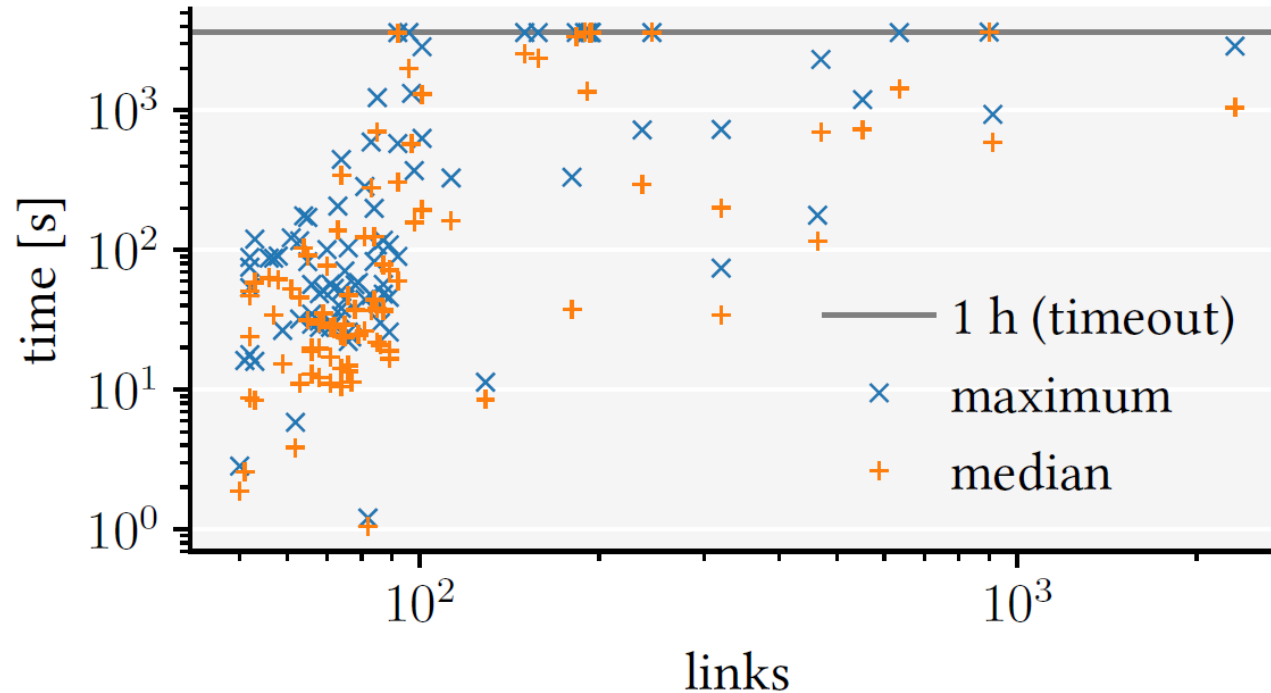


Efficient

*Few minutes for  
hundreds of links*

Waypoint, *four 9s* precision

# Single-flow Properties



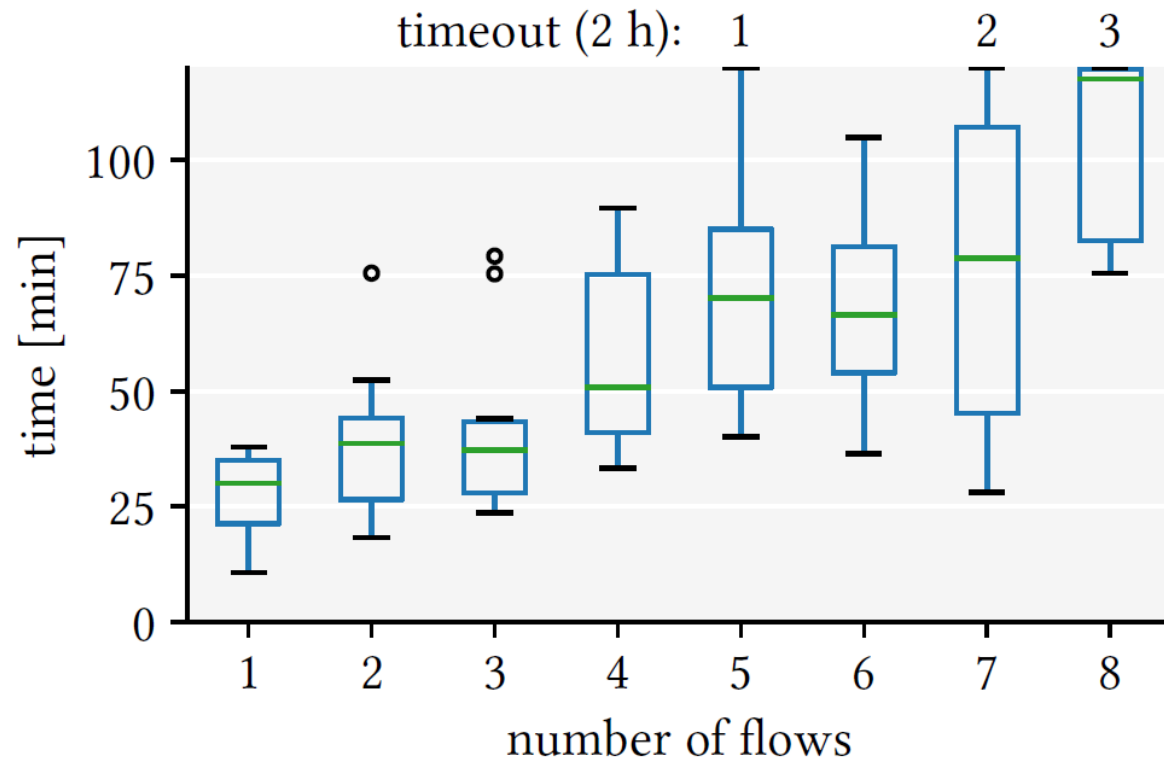
Efficient

*Few minutes for  
hundreds of links*

For 80% of scenarios,  
> 50% of links are ❄️

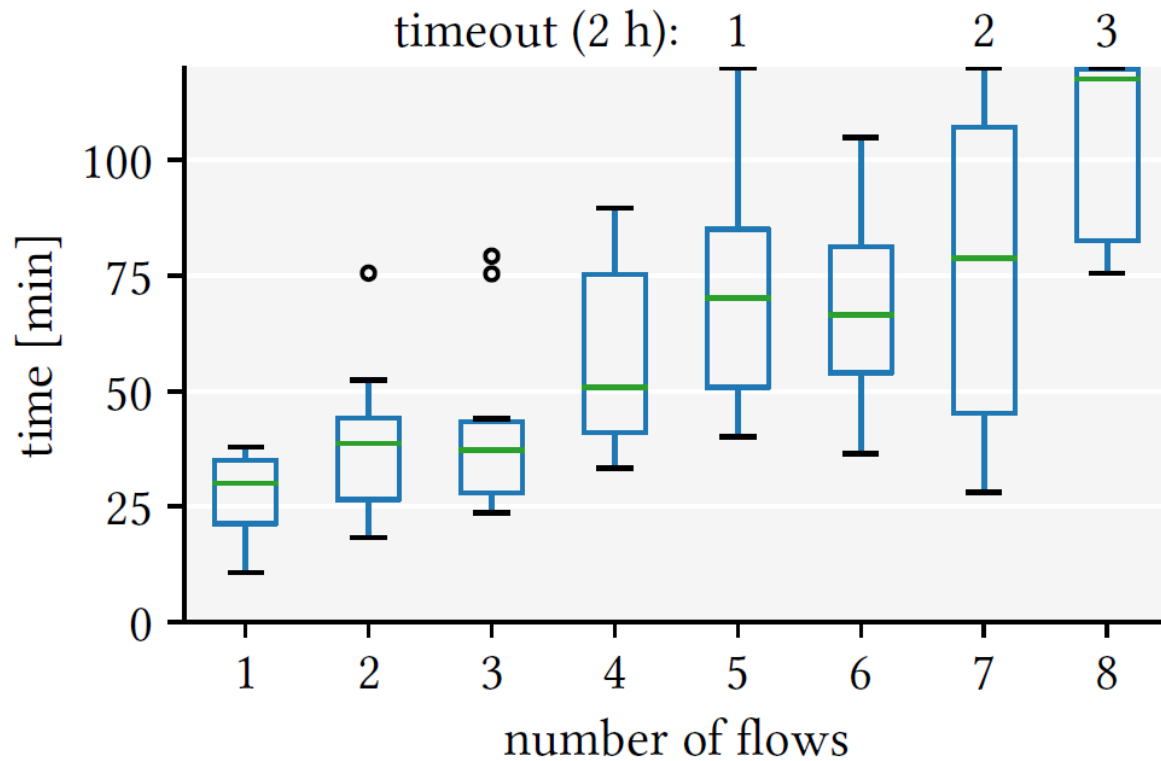
Waypoint, *four 9s* precision

# More flows



Congestion, 235 links network

# More flows

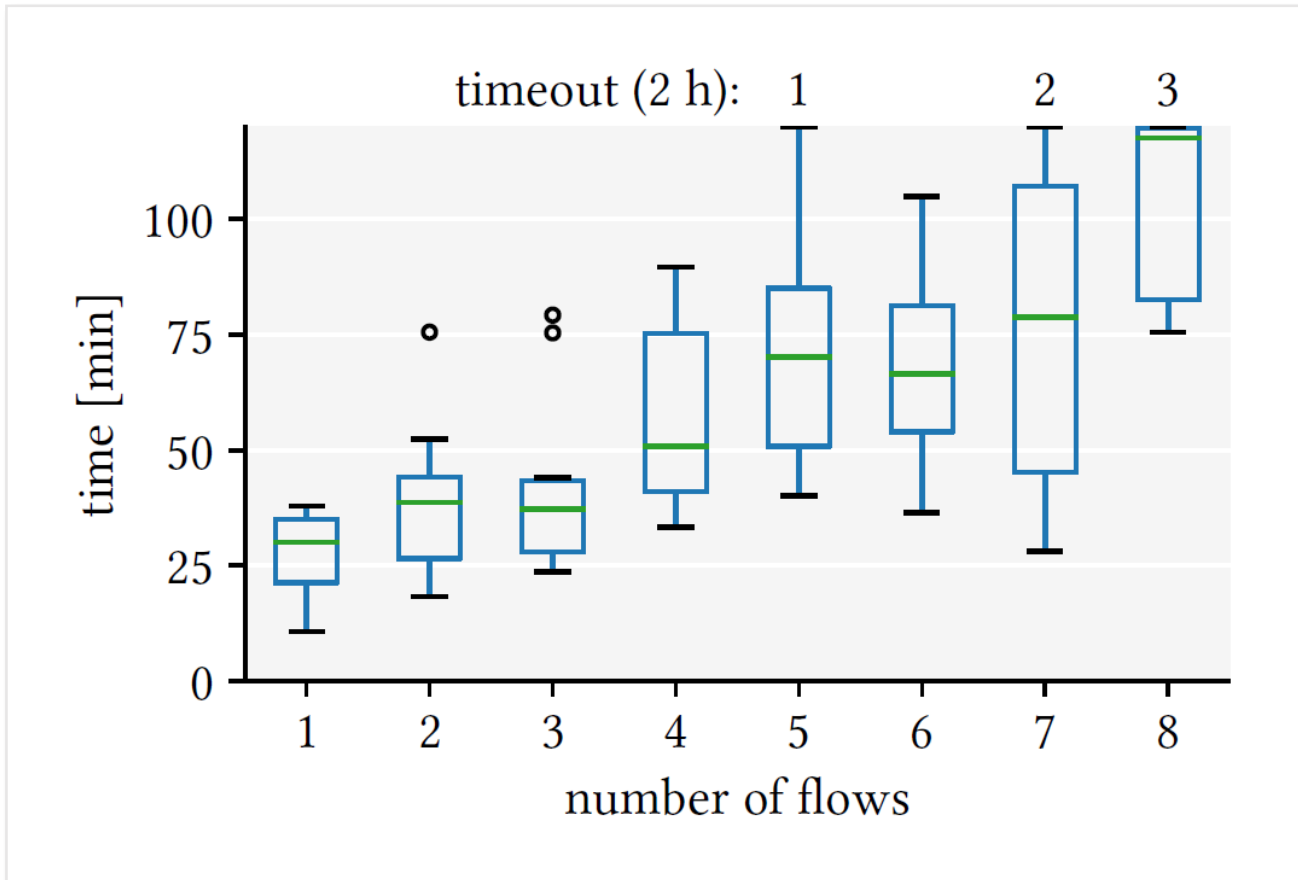


Efficient for few flows

Performance degrades gracefully

Congestion, 235 links network

# More flows



Efficient for few flows

Performance degrades gracefully

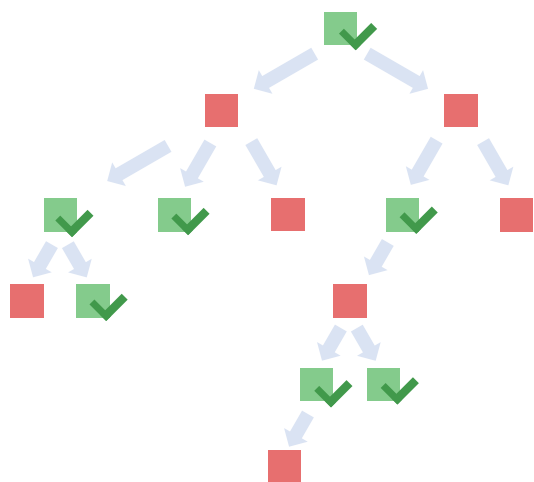
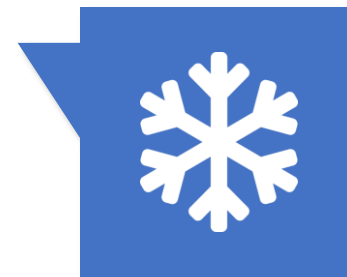
See paper for more...

Congestion, 235 links network

# Summary



Cold edges



nsg-ethz/netdice

BGP + IGP

High accuracy

# Icon Credits

Icons by FontAwesome (CC BY 4.0), <https://fontawesome.com/license>