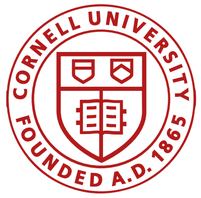
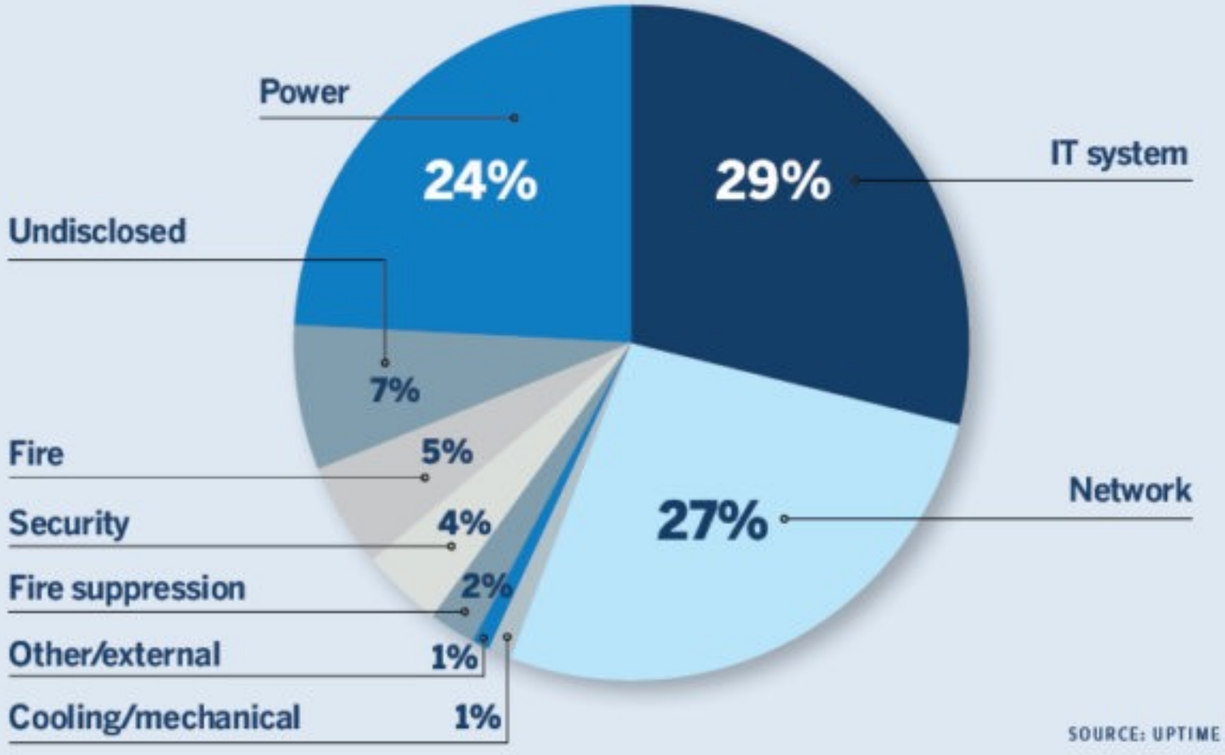
Scalable Verification of Probabilistic Networks

Steffen Smolka, Praveen Kumar, David Kahn,
Nate Foster, Justin Hsu, Dexter Kozen, Alexandra Silva



DATA CENTER OUTAGES

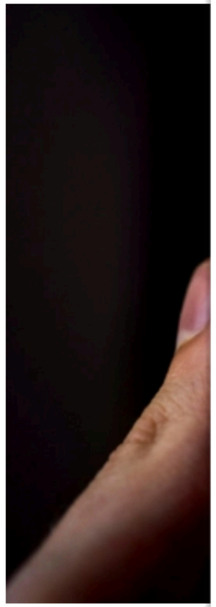
Network and IT systems were the leading causes of publicly recorded data center outages between 2016 and 2018, according to Uptime Institute.



SOURCE: UPTIME INSTITUTE

HOME / BUSINESS / BREAKING NEWS

Instagram Outage Follows Disruption To PlayStation Network – Update



Valentin WolValen

A CENTE
and IT systems
tages between

The New York Times

Google Disruptions Affect Gmail, YouTube and Other Sites

Errors and slowdowns affected a number of Google services on Sunday. The company said they were the result of "high levels of network congestion."

Christie Hemm Klok for The New York Times

WIRED SUBSCRIBE

BRIAN BARRETT SECURITY 03.13.19 07:18 PM

WHEN FACEBOOK GOES DOWN, DON'T BLAME HACKERS

DEADLINE TIP US

HOME / BUSINESS / BREAKING NEWS

Gmail Goes Down In Second Sizable Outage To Hit Google This Month

By Dade Hayes

June 17, 2019 12:53pm

HOME / BUSINESS / BREAKING NEWS

Instagram Outage Follows Disruption To PlayStation Network – Update

CENTE
and IT systems
tages between

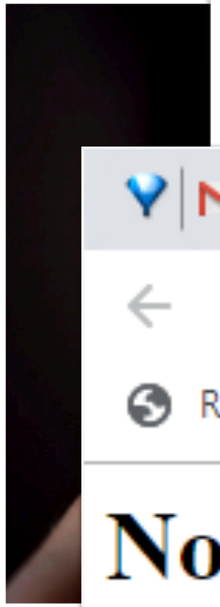
WIRED SUBSCRIBE

BRIAN BARRETT SECURITY 03.13.19 07:18 PM

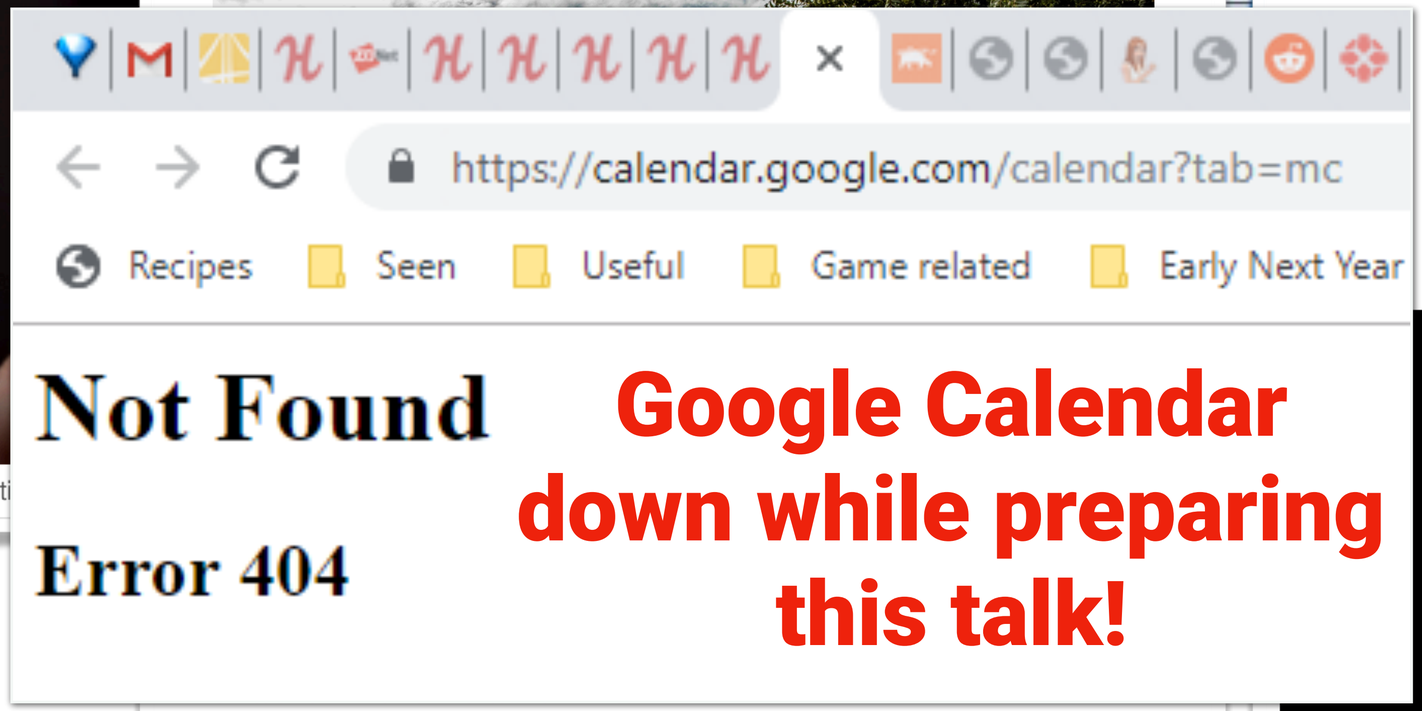
WHEN FACEBOOK GOES DOWN, DON'T BLAME HACKERS

The New York Times

Google Disruptions Affect Gmail, YouTube and Other Sites



Valenti



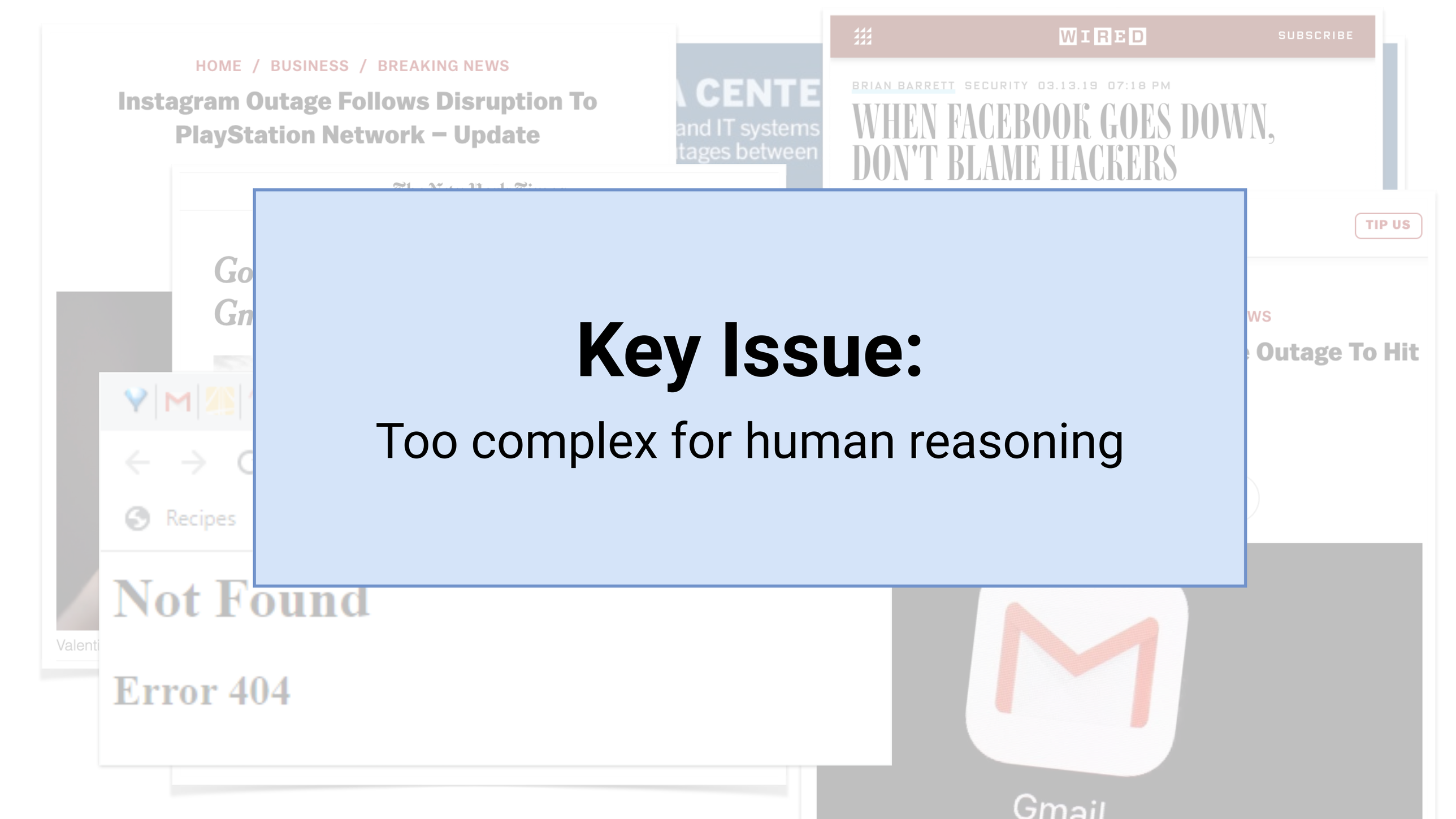
Google Calendar down while preparing this talk!

DEADLINE TIP US

HOME / BUSINESS / BREAKING NEWS

Gmail Goes Down In Second Sizable Outage To Hit Google This Month

By Dade Hayes
June 17, 2019 12:53pm

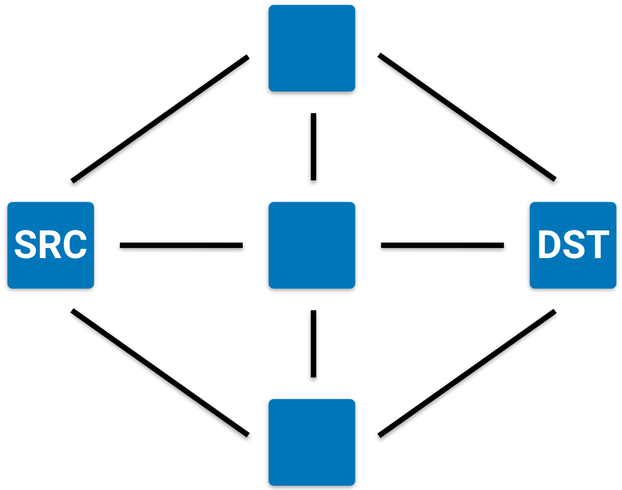


Key Issue:

Too complex for human reasoning

Example: Network Verification

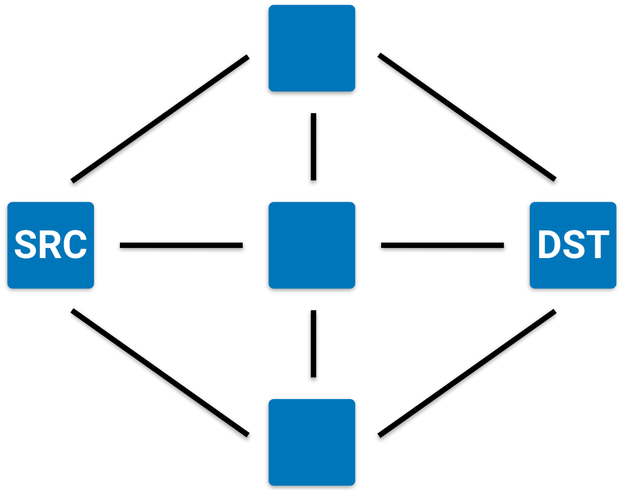
Network topology:



Network config:
shortest path

Example: Network Verification

Network topology:

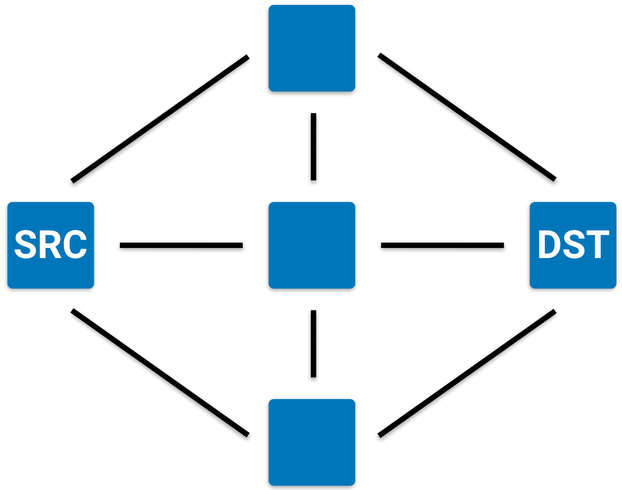


Network config:
shortest path

Verification Tool

Example: Network Verification

Network topology:



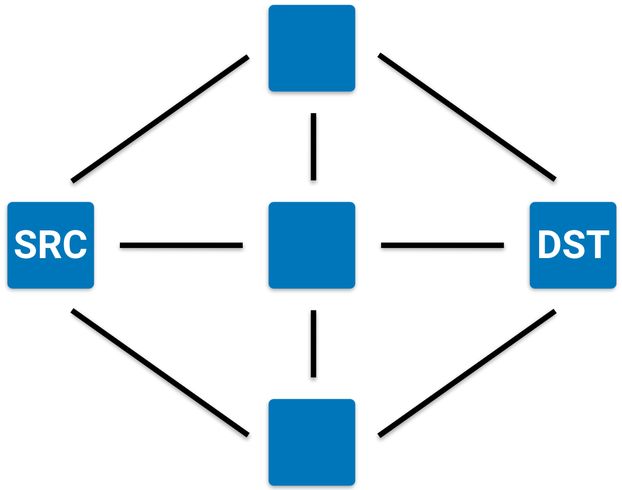
“Will my packet get delivered?

Verification Tool

Network config:
shortest path

Example: Network Verification

Network topology:



"Will my packet get delivered?

Verification Tool

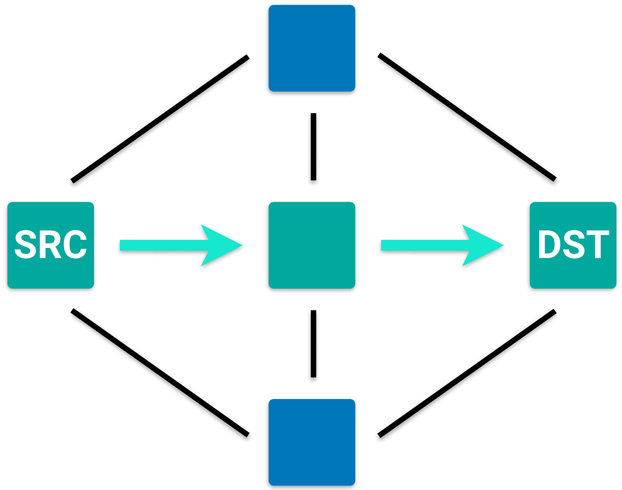
No!
(+ counterexample)

Yes!
(+ formal proof)

Network config:
shortest path

Example: Network Verification

Network topology:



“Will my packet get delivered?

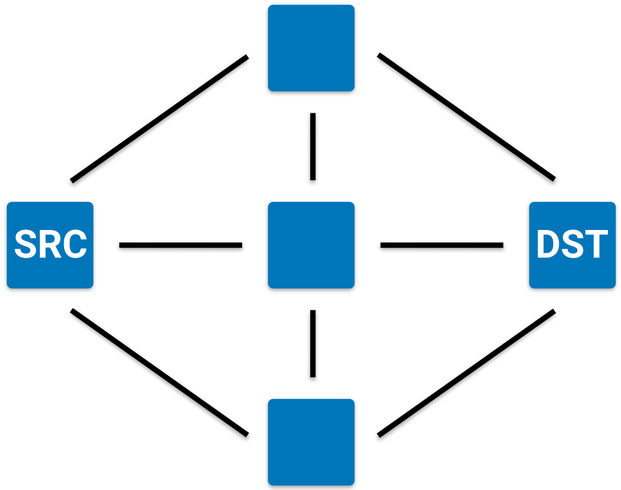
Verification Tool

Network config:
shortest path

Yes!
(+ formal proof)

Example: Network Verification

Network topology:



“Will my packet get delivered?

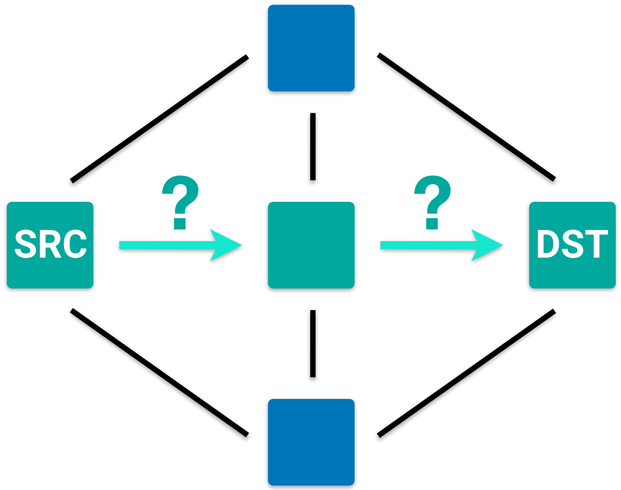
Verification Tool

Network config:
shortest path

Failure model:
links fail independently with probability 1%

Example: Network Verification

Network topology:



"Will my packet get delivered?

Verification Tool

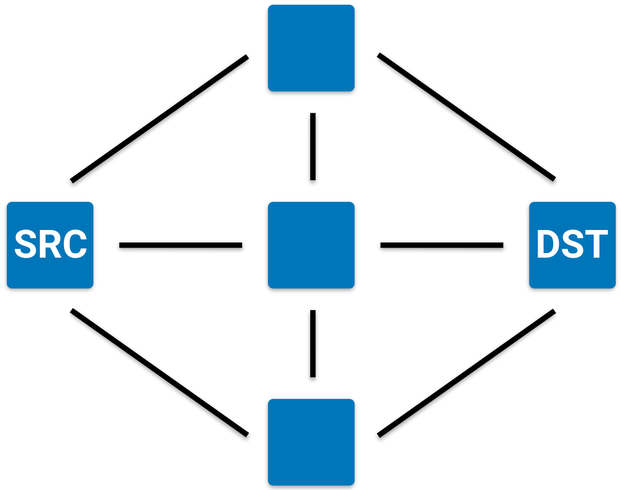
maybe

Network config:
shortest path

Failure model:
links fail independently with probability 1%

Example: Probabilistic Network Verification

Network topology:



“Will my packet get delivered?

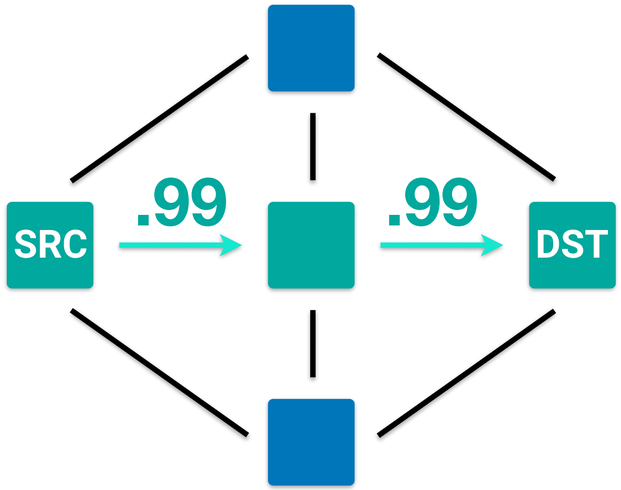
McNetKAT

Network config:
shortest path

Failure model:
links fail independently with probability 1%

Example: Probabilistic Network Verification

Network topology:



"Will my packet get delivered?

McNetKAT

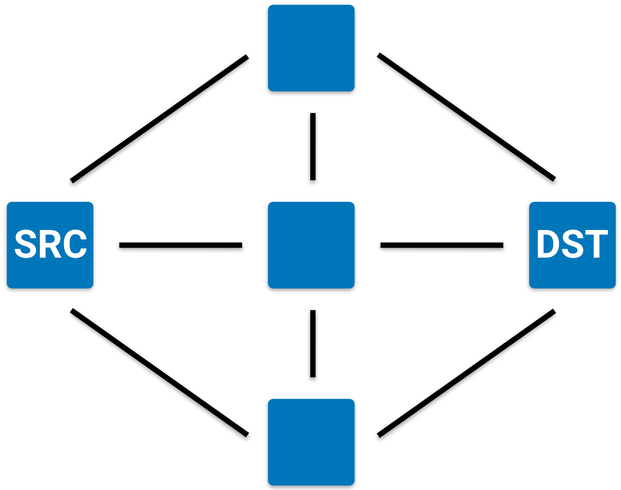
with probability $(0.99)^2$

Network config:
shortest path

Failure model:
links fail independently with probability 1%

Example: Probabilistic Network Verification

Network topology:



“Will my packet get delivered?

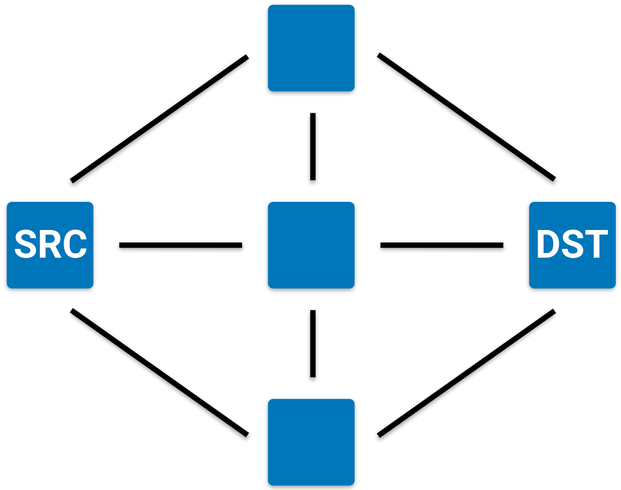
McNetKAT

Network config:
shortest path
+ detour uniform at random

Failure model:
≤ 2 links fail independently
with probability 1%

Example: Probabilistic Network Verification

Network topology:



“Will my packet get delivered?

Network config:
shortest path
+ detour uniform at random

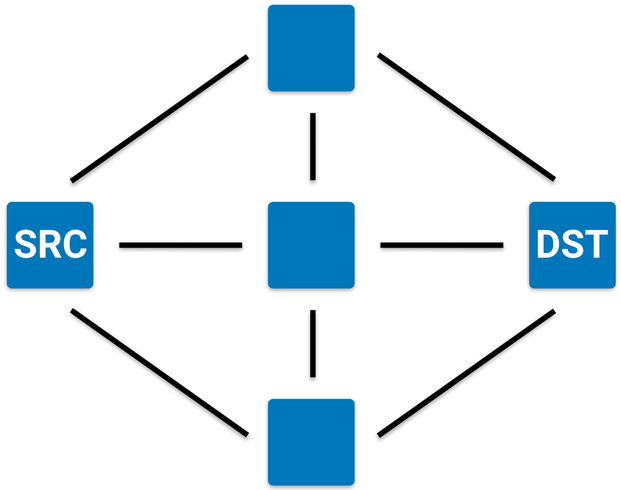
Failure model:
≤ 2 links fail independently
with probability 1%

McNetKAT

with probability 1

Example: Probabilistic Network Verification

Network topology:



“Will my packet get delivered?

Network config:
shortest path
+ detour uniform at random

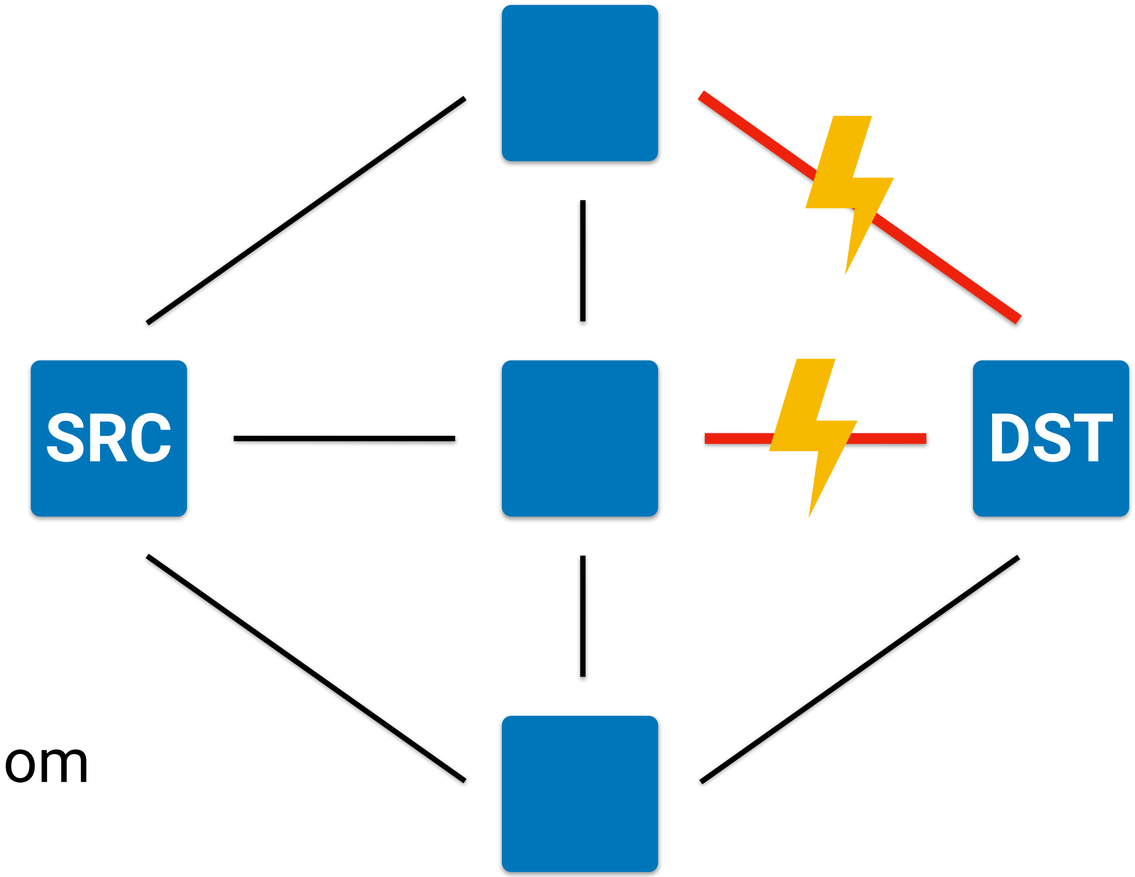
Failure model:
≤ 2 links fail independently
with probability 1%

McNetKAT

with probability 1

→ configuration is **2-resilient**

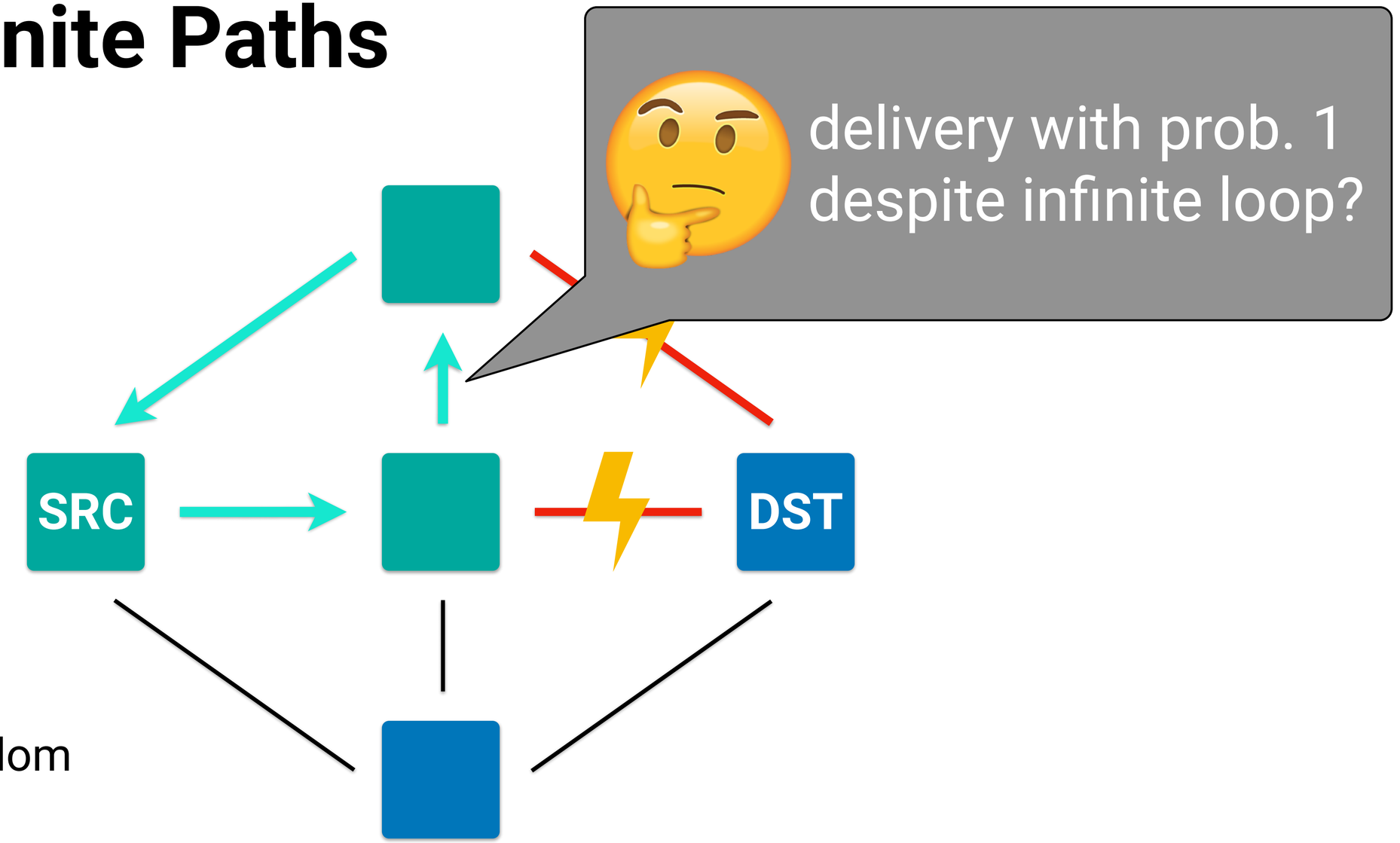
Example: Infinite Paths



Network config:
shortest path
+ detour uniform at random

Failure model:
≤ 2 links fail independently
with probability 1%

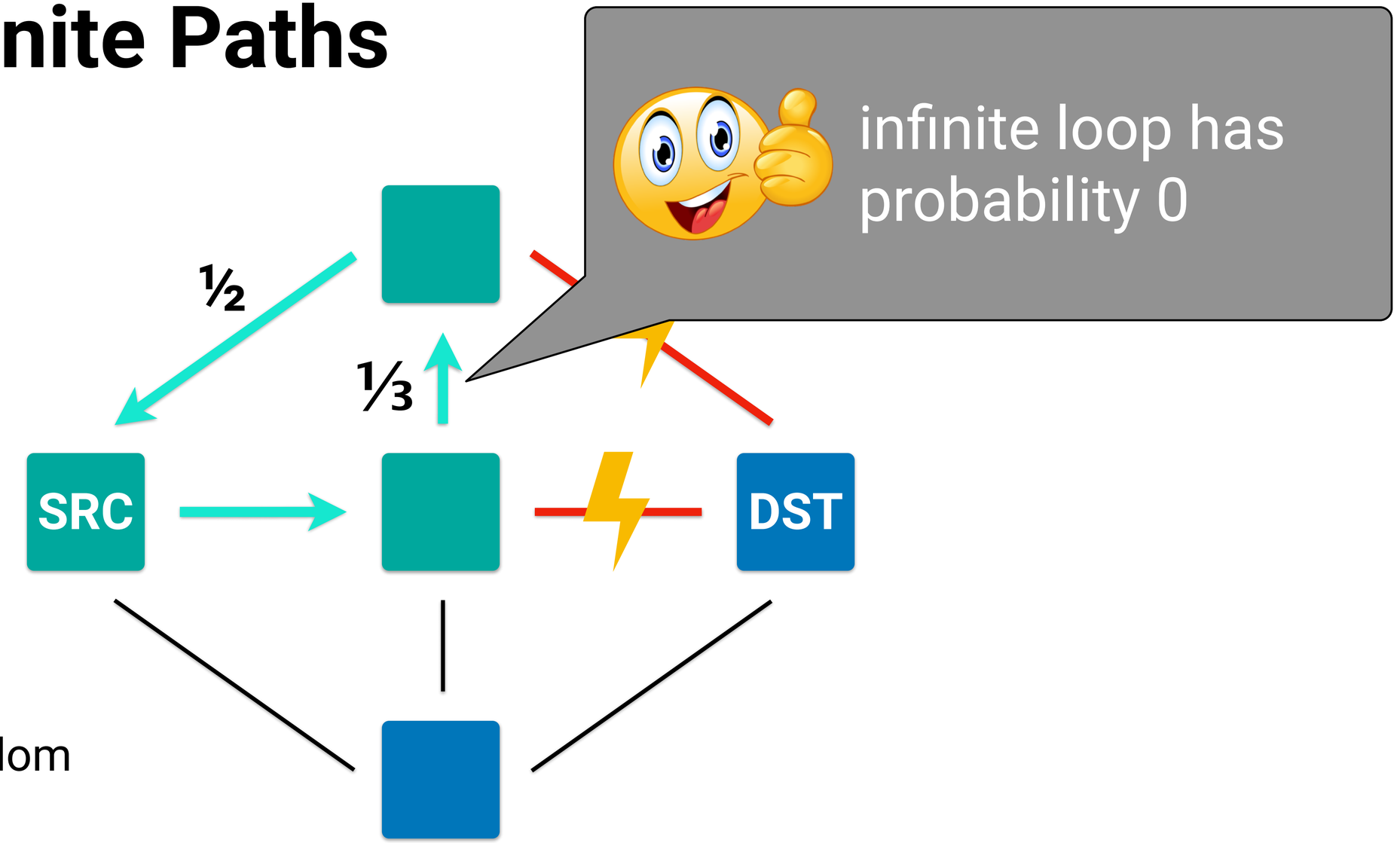
Example: Infinite Paths



Network config:
shortest path
+ detour uniform at random

Failure model:
≤ 2 links fail independently
with probability 1%

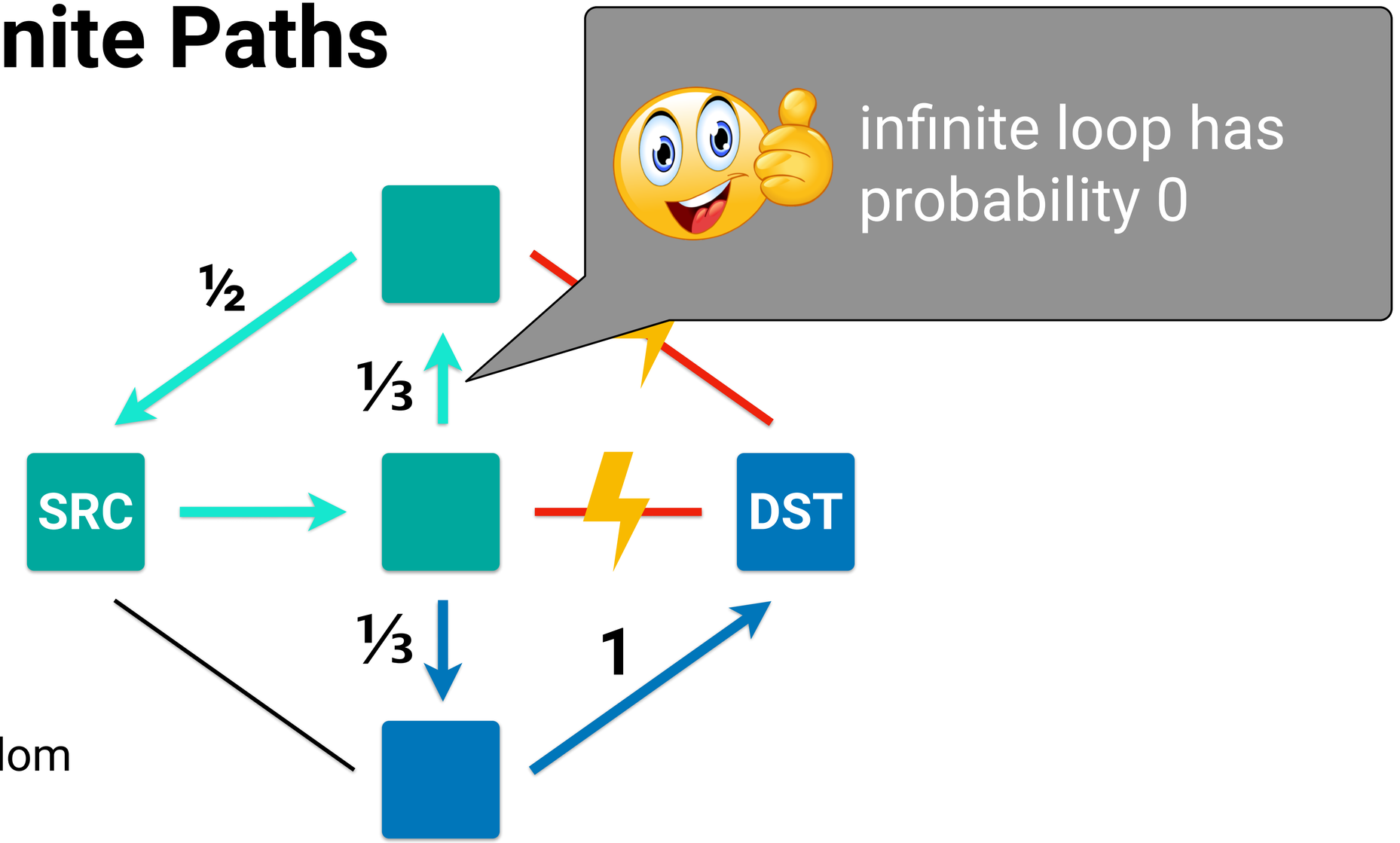
Example: Infinite Paths



Network config:
shortest path
+ detour uniform at random

Failure model:
≤ 2 links fail independently
with probability 1%

Example: Infinite Paths



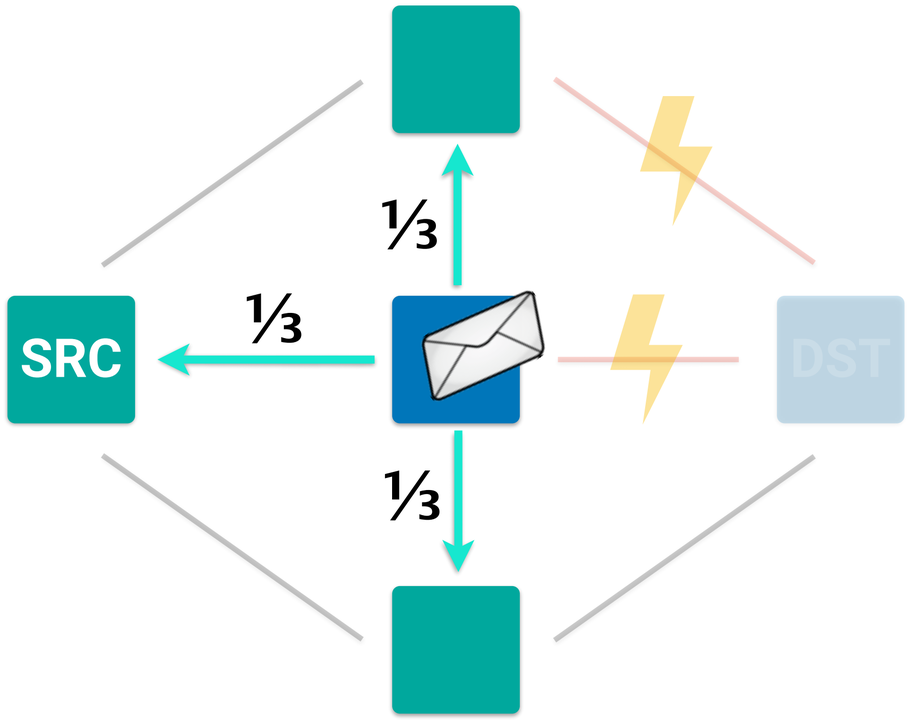
Network config:
shortest path
+ detour uniform at random

Failure model:
≤ 2 links fail independently
with probability 1%

Challenge:

How to automate this?

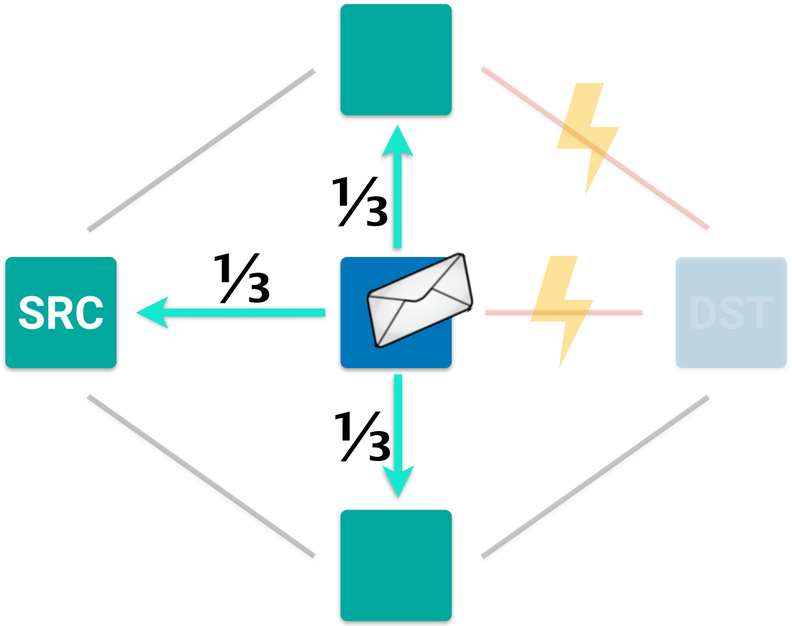
Solution



Packets in probabilistic network undergo **random walk**

- can be modeled as **Markov chain**
- limiting distribution computable in **closed form**

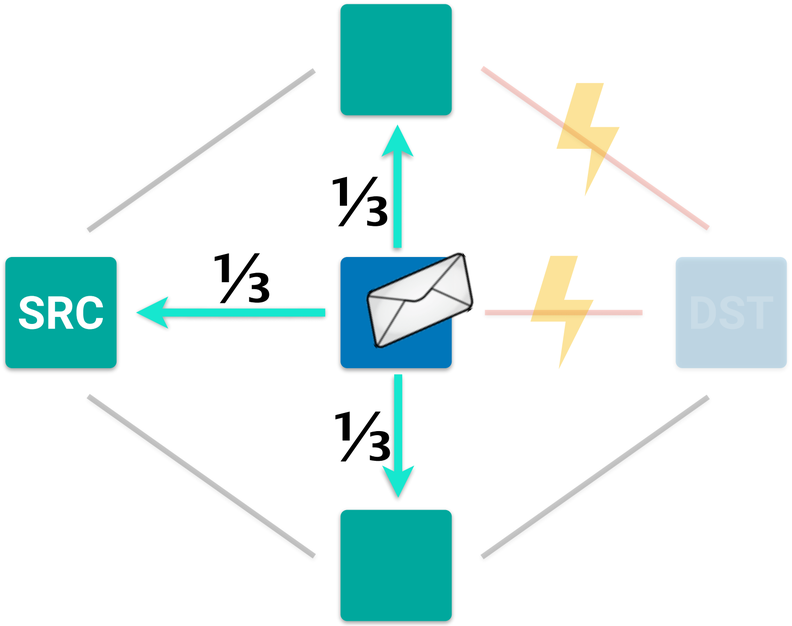
Challenge: State Space



State Space

Location x Header Values

Challenge: State Space

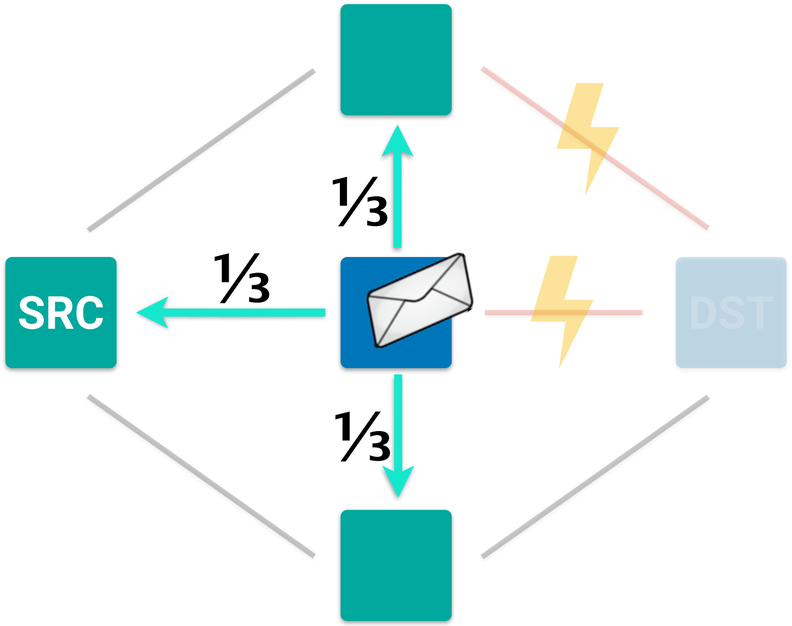


State Space

Location x Header Values

(~1k) (~ 2^{160})

Challenge: State Space



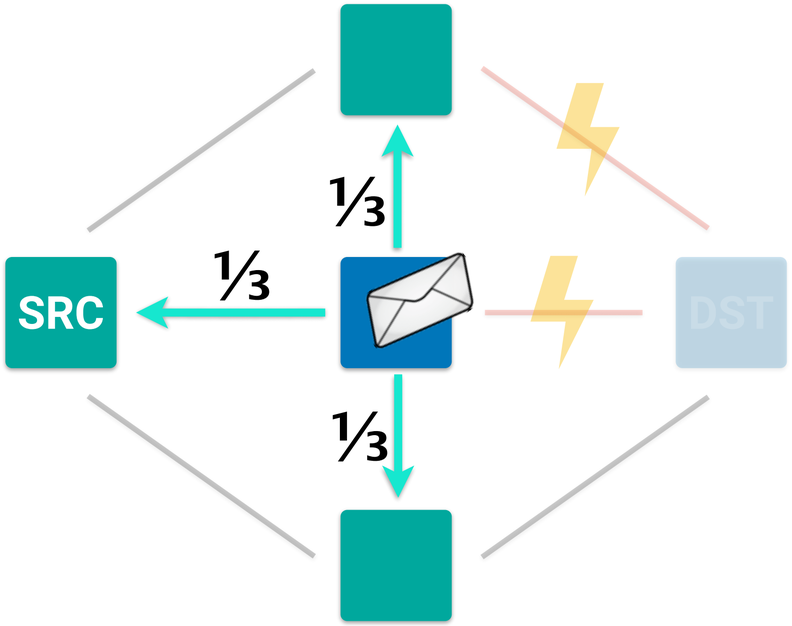
State Space

Location x Header Values

(~1k) (~ 2^{160})

larger than # particles in the universe

Challenge: State Space



State Space

Location x Header Values

(~1k) (~ 2^{160})

But: Markov chain very structured in practice

- sparse transition structure
- many similar states
- analysis can be made **tractable** using clever data structures

Implementation

McNetKAT architecture

```
if port=1 then
  port←2 ⊕0.5 port←3
else if port=2 then
  port←1
else
  drop
```

network model

= ProbNetKAT program

[ESOP ’16, POPL ’17]

McNetKAT architecture

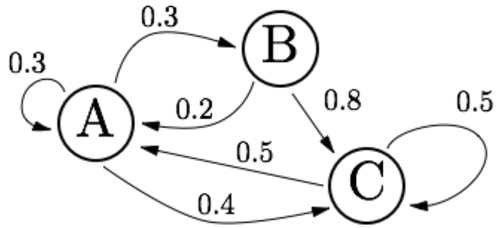
```
if port=1 then
  port←2 ⊕0.5 port←3
else if port=2 then
  port←1
else
  drop
```

network model

= ProbNetKAT program

[ESOP '16, POPL '17]

⟦-⟧



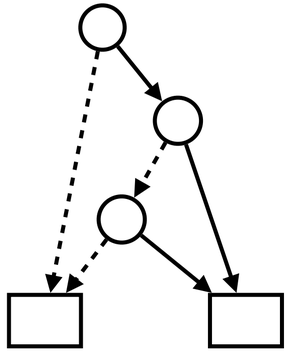
Markov chain

McNetKAT architecture

```
if port=1 then
  port←2 ⊕0.5 port←3
else if port=2 then
  port←1
else
  drop
```

network model
= ProbNetKAT program
[ESOP '16, POPL '17]

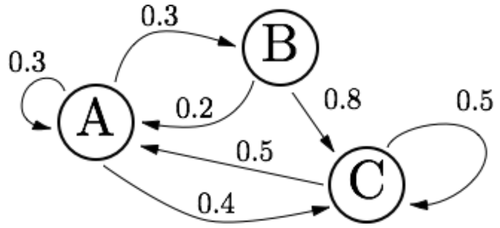
compile



symbolic IR
[ICFP '15]

⟦-⟧

⟦-⟧



Markov chain

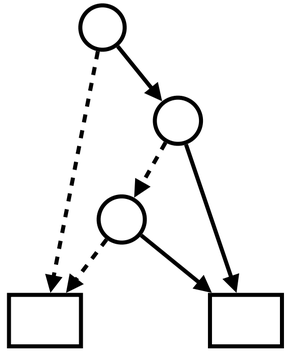
McNetKAT architecture

compute limiting distribution

```
if port=1 then
  port←2 ⊕0.5 port←3
else if port=2 then
  port←1
else
  drop
```

network model
= ProbNetKAT program
[ESOP '16, POPL '17]

compile



symbolic IR
[ICFP '15]

convert

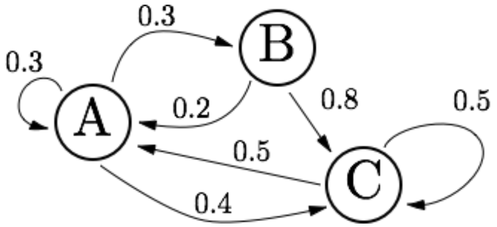
$$\begin{bmatrix} 1 & & & & \\ & & \frac{1}{2} & \frac{1}{2} & \\ & 1 & & & \\ & 1 & & & \\ 1 & & & & \end{bmatrix}$$

sparse matrix

⟦-⟧

⟦-⟧

⟦-⟧



Markov chain

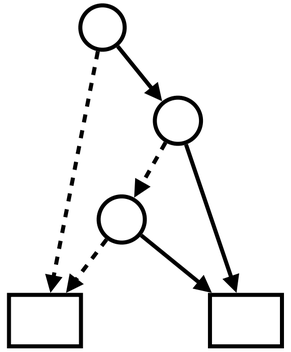
McNetKAT architecture

compute limiting distribution

```
if port=1 then
  port←2 ⊕0.5 port←3
else if port=2 then
  port←1
else
  drop
```

network model
= ProbNetKAT program
[ESOP '16, POPL '17]

compile



symbolic IR
[ICFP '15]

convert

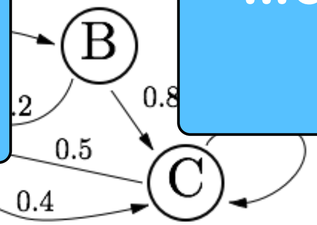
$$\begin{bmatrix} 1 & & & & \\ & & \frac{1}{2} & \frac{1}{2} & \\ & 1 & & & \\ & 1 & & & \\ 1 & & & & \end{bmatrix}$$

sparse matrix

[[-]]

can be analyzed using custom code…

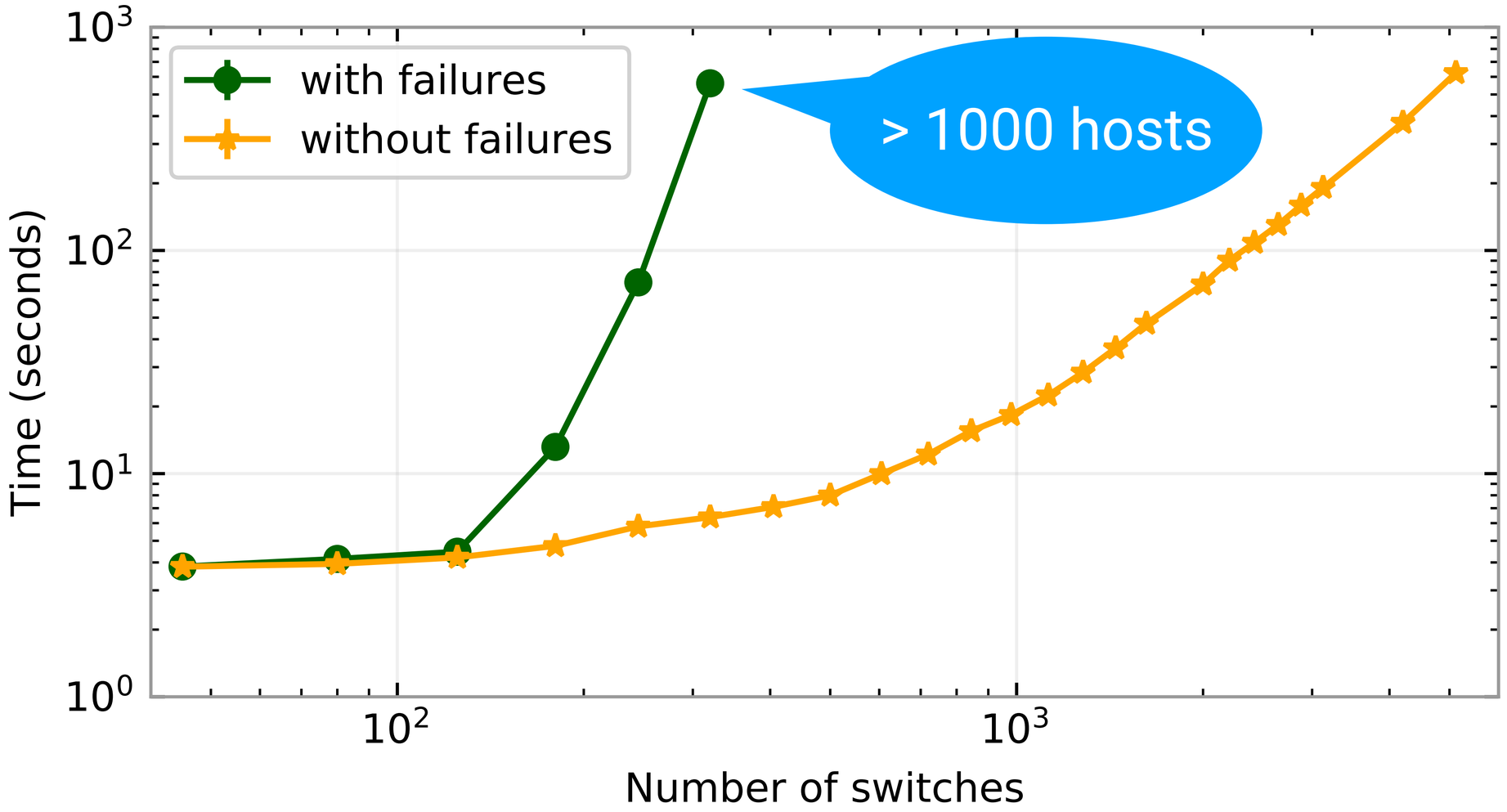
…and standard tools



Markov chain

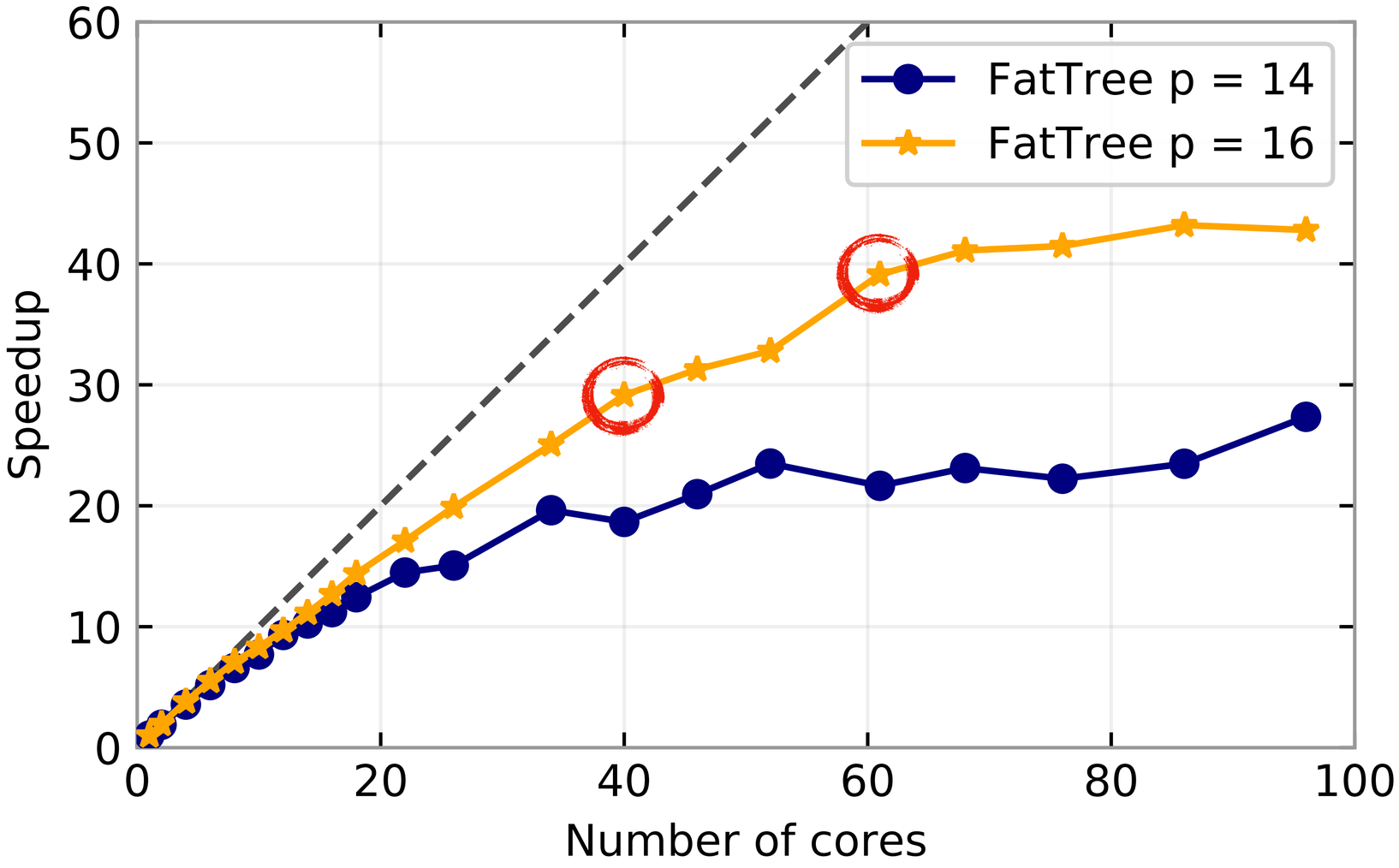
Evaluation

Scalability on FatTree with ECMP

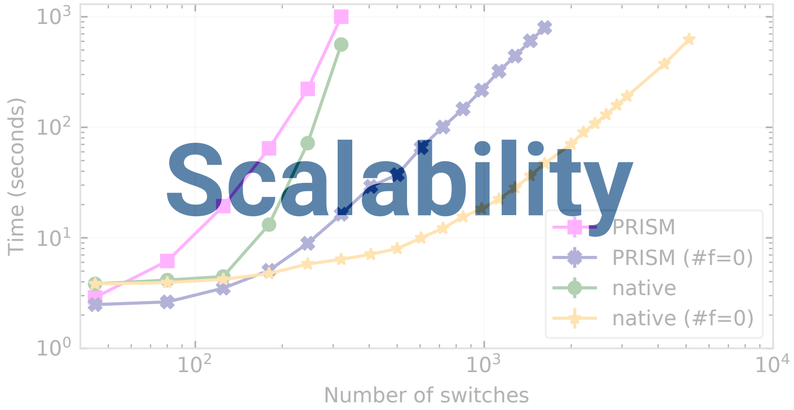


McNetKAT scales to data-center-size networks.

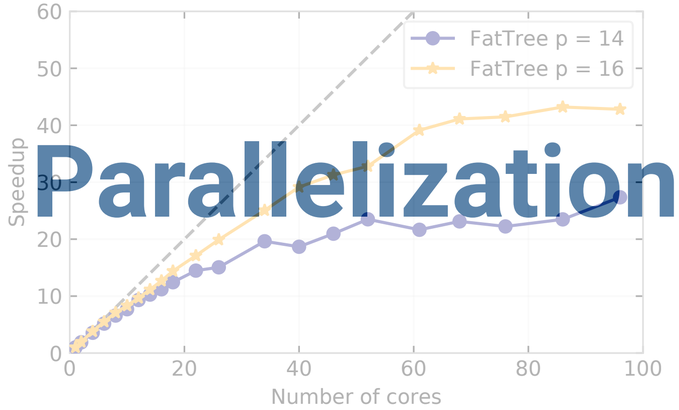
Speedup through parallelization



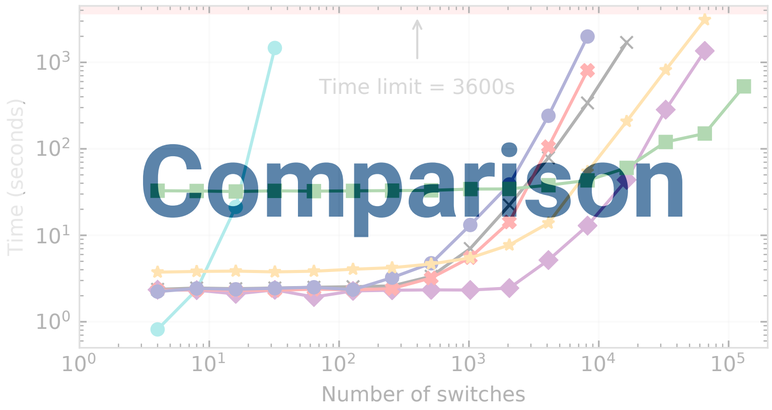
Often near optimal speedup. Up to 40x on 60 cores.



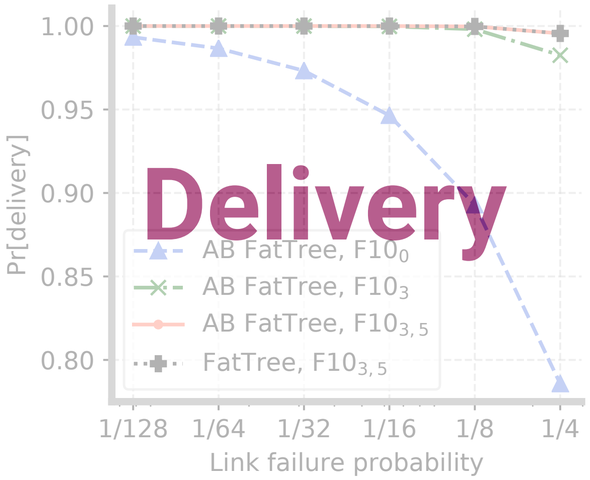
Scalability



Parallelization



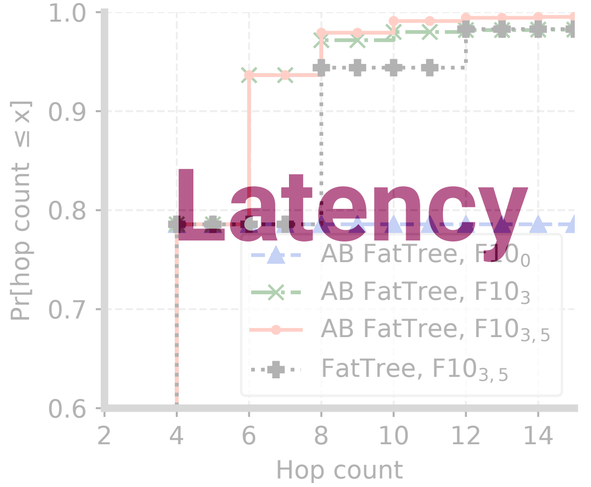
Comparison



Delivery

k	$\widehat{\mathrm{M}}(\mathrm{F10}_0, f_k) \equiv$ *teleport*	$\widehat{\mathrm{M}}(\mathrm{F10}_3, f_k) \equiv$ *teleport*	$\widehat{\mathrm{M}}(\mathrm{F10}_{3,5}, f_k) \equiv$ *teleport*
0	✓	✓	✓
1	✗	✓	✓
2	✗	✓	✓
3	✗	✗	✓
4	✗	✗	✗
∞	✗	✗	✗

Resilience



Latency

Summary: McNetKAT

First **scalable** verification tool for **probabilistic** networks

- can reason, e.g., about fault-tolerance

Based on theory of Markov chains

- provides solid mathematical foundation
- enables computing limits in closed form

Scales thanks to

- sparsity, symbolic data structures
- parallelization
- optimized linear algebra solver

Thank you!

Code available: https://smolka.st/artifacts/mcnetkat/

Comparison vs state of the art