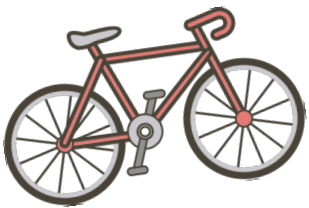
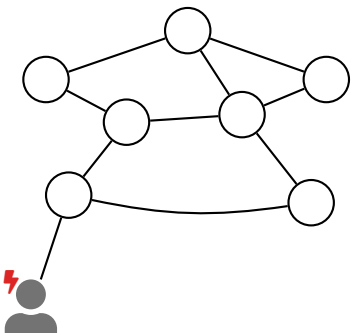
Verifying maximum link loads in a changing world

Tibor Schneider[1], Stefano Vissicchio[2], Laurent Vanbever[1]

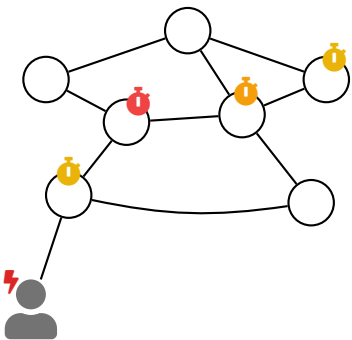
NSDI, April 30, 2025



1 ETH zürich
2 UCL

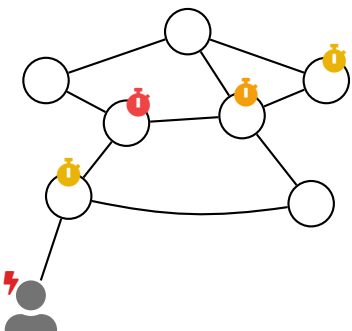


Link loads are a key indicator for network performance.



Monitor the live network.

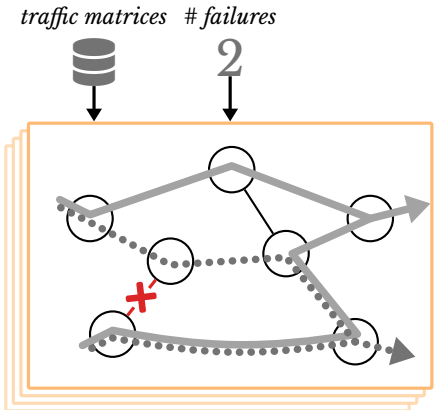
Link loads are a key indicator for network performance.



Monitor the live network.

- Can it happen again *in the future*?

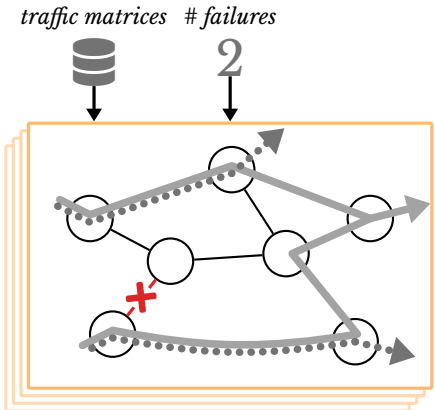
Recent systems find worst-case link loads under failures.



Find *worst-case* loads under arbitrary failures.

[1] Kausik Subramanian et al. “Detecting network load violations for distributed control planes”. In: *ACM SIGPLAN*. 2020
[2] Ruihan Li et al. “A General and Efficient Approach to Verifying Traffic Load Properties under Arbitrary k Failures”. In: *ACM SIGCOMM*. 2024

Recent systems find worst-case link loads under failures.

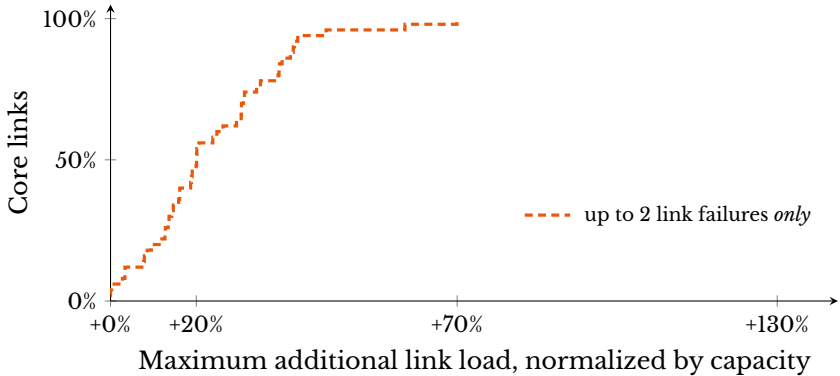


Find *worst-case* loads under arbitrary failures.

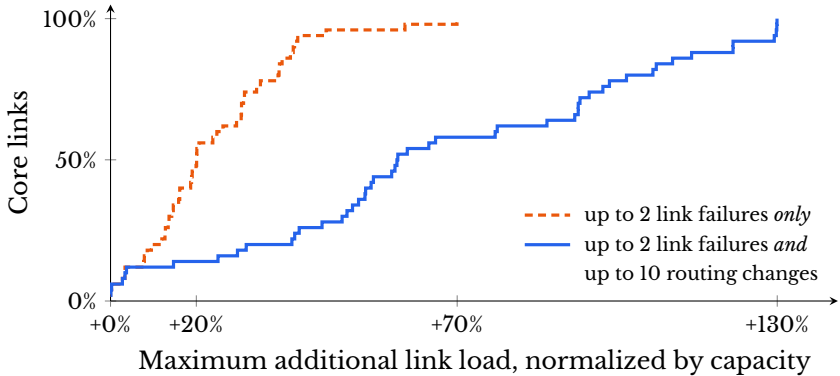
- Traffic also depends on *BGP routing inputs.*

[1] Kausik Subramanian et al. "Detecting network load violations for distributed control planes". In: *ACM SIGPLAN*. 2020
[2] Ruihan Li et al. "A General and Efficient Approach to Verifying Traffic Load Properties under Arbitrary k Failures". In: *ACM SIGCOMM*. 2024

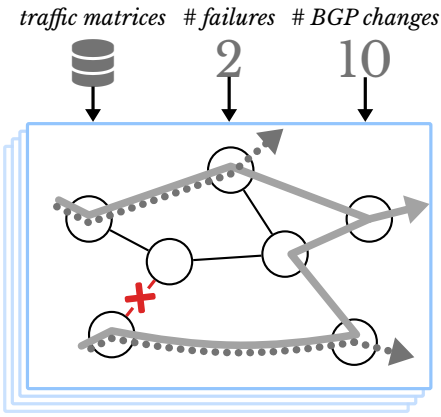
Ignoring routing changes leads to underestimating worst-case loads.



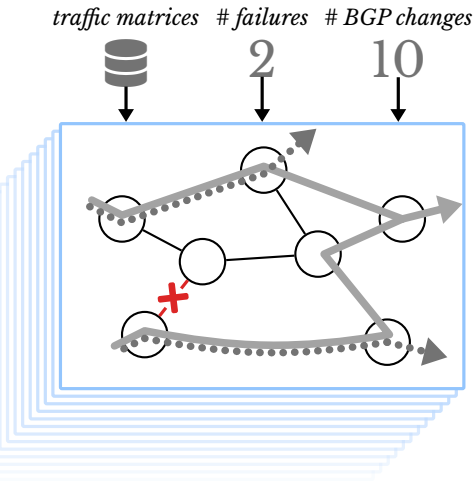
Ignoring routing changes leads to underestimating worst-case loads.



The space of failures *and* route changes is huge and difficult to navigate.

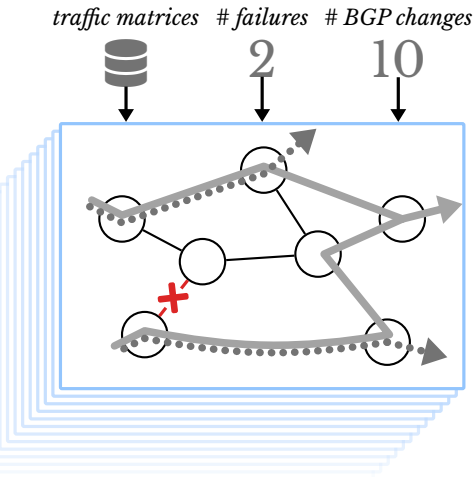


The space of failures *and* route changes is huge and difficult to navigate.



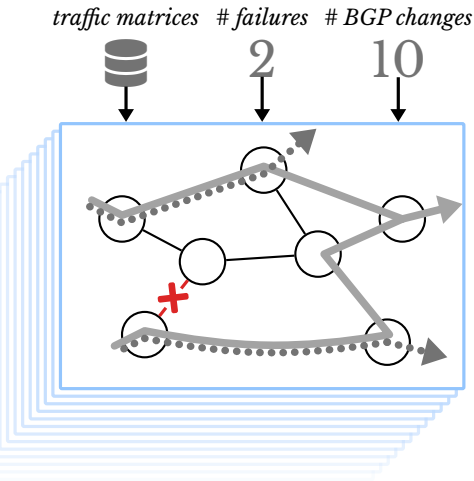
- A destination can be advertised by *any subset* of BGP neighbors.

The space of failures *and* route changes is huge and difficult to navigate.



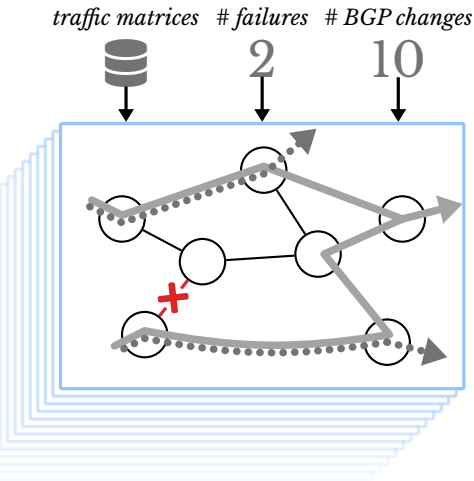
- A destination can be advertised by *any subset* of BGP neighbors.
- Failures create *dependencies* between destinations.

The space of failures *and* route changes is huge and difficult to navigate.



- A destination can be advertised by *any subset* of BGP neighbors.
- Failures create *dependencies* between destinations.
- Over *one million* of destination prefixes.

The space of failures *and* route changes is huge and difficult to navigate.



- A destination can be advertised by *any subset* of BGP neighbors.
- Failures create *dependencies* between destinations.
- Over *one million* of destination prefixes.

Velo: **Ve**rify maximum link **lo**ads under failures and routing changes

Search space reduction:

A single egress router maximizes link loads.

Input size reduction:

Cluster destination with similar traffic patterns.

Velo: **Ve**rify maximum link **lo**ads under failures and routing changes

Search space reduction:

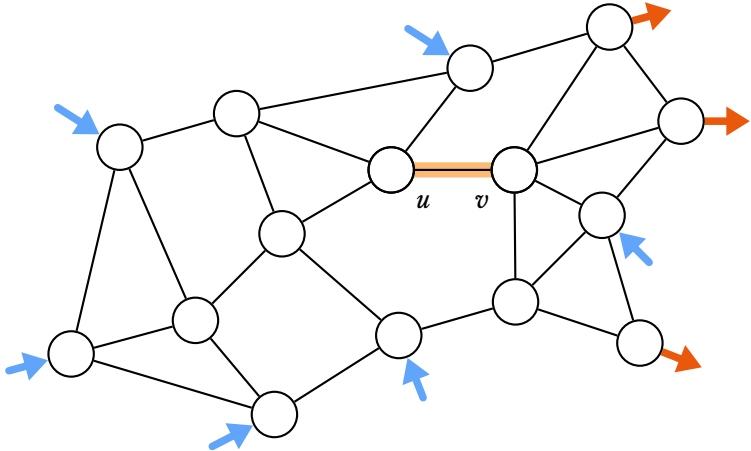
A single egress router maximizes link loads.

Input size reduction:

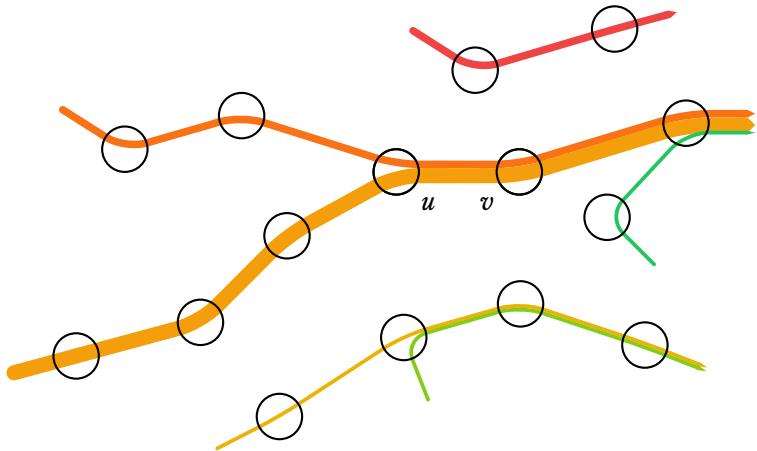
Cluster destination with similar traffic patterns.

Idea: Consider each failure separately.

- In a given failure scenario,
  worst-case states for destinations are *independently*.

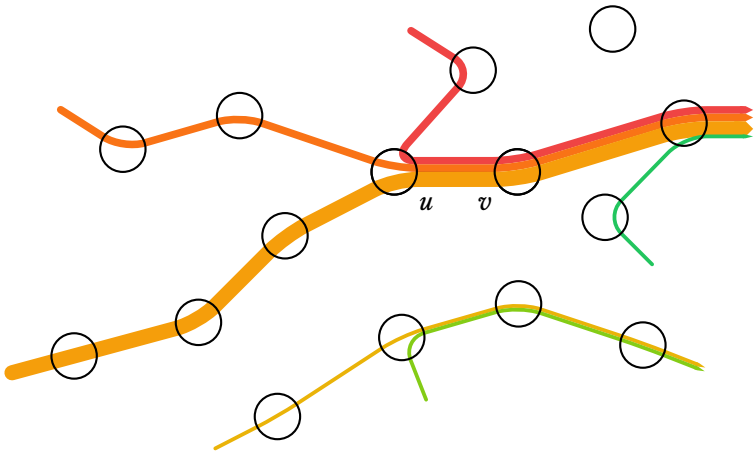

u
v

Nodes forwarding traffic via u–v choose the same route as v.

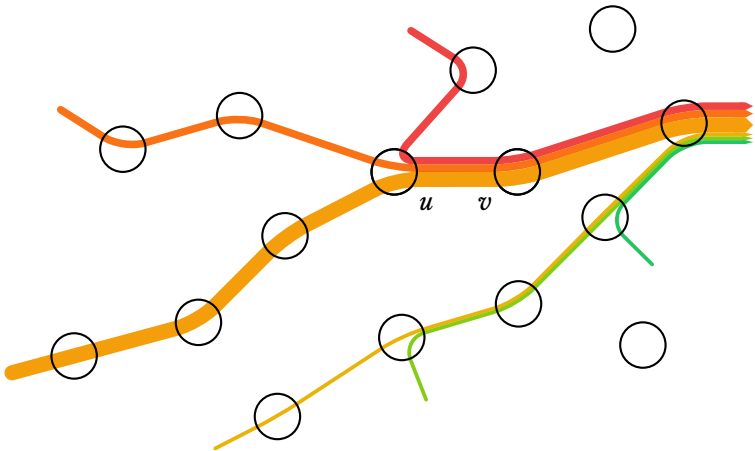


Nodes forwarding traffic via u–v choose the same route as v.

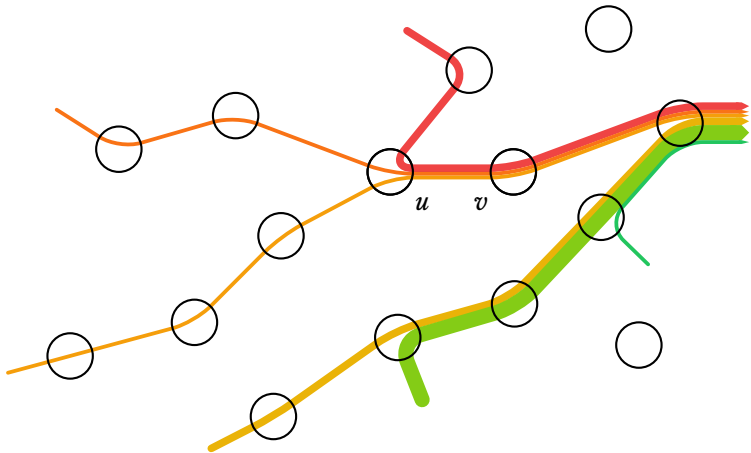
Traffic on u–v cannot decrease when removing egresses not used by v.



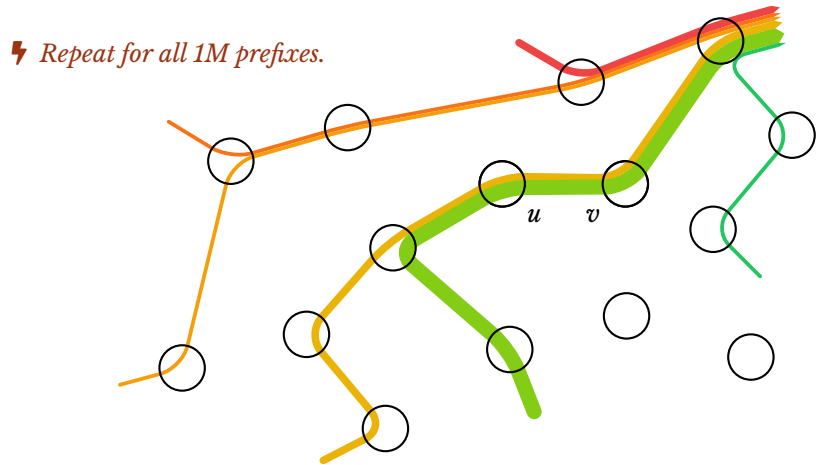
Nodes forwarding traffic via u–v choose the same route as v.
Traffic on u–v cannot decrease when removing egresses not used by v.
Hence, worst-case load is achieved with a *single egress*.



Nodes forwarding traffic via u–v choose the same route as v.
Traffic on u–v cannot decrease when removing egresses not used by v.
Hence, worst-case load is achieved with a *single egress*.



Nodes forwarding traffic via u–v choose the same route as v.
Traffic on u–v cannot decrease when removing egresses not used by v.
Hence, worst-case load is achieved with a *single egress*.



Velo: **Ve**rify maximum link **lo**ads under failures and routing changes

Search space reduction:

A single egress router maximizes link loads.

Input size reduction:

Cluster destination with similar traffic patterns.

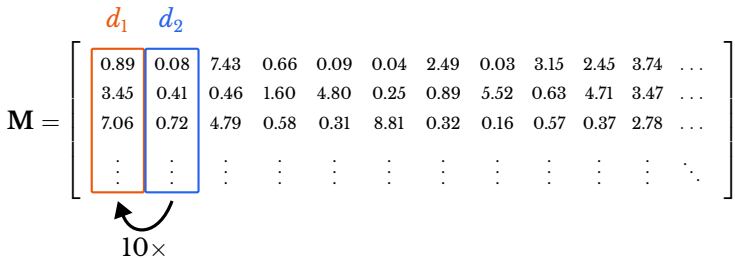
We can “combine” traffic for destinations with *similar distributions*.

$$\mathbf{M} = \begin{bmatrix} 0.89 & 0.08 & 7.43 & 0.66 & 0.09 & 0.04 & 2.49 & 0.03 & 3.15 & 2.45 & 3.74 & \dots \\ 3.45 & 0.41 & 0.46 & 1.60 & 4.80 & 0.25 & 0.89 & 5.52 & 0.63 & 4.71 & 3.47 & \dots \\ 7.06 & 0.72 & 4.79 & 0.58 & 0.31 & 8.81 & 0.32 & 0.16 & 0.57 & 0.37 & 2.78 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

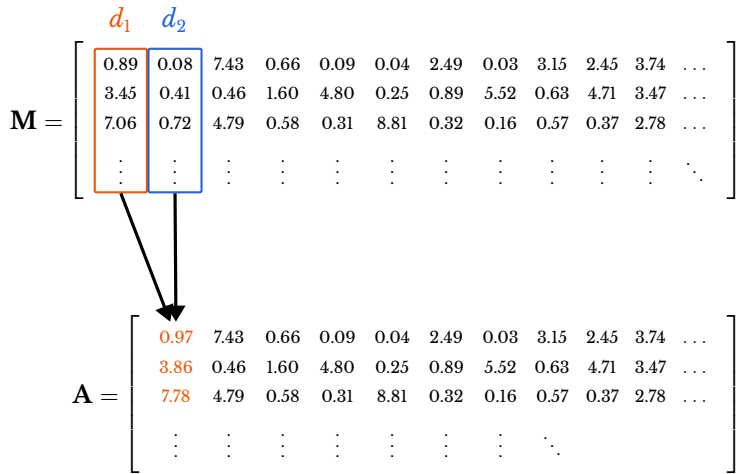
ingress points

destination prefixes

We can “combine” traffic for destinations with *similar distributions*.



We can “combine” traffic for destinations with *similar distributions*.



Velo approximates the traffic matrix by clustering destinations.

$$\mathbf{M} = \begin{bmatrix} 0.89 & 0.08 & 7.43 & 0.66 & 0.09 & 0.04 & 2.49 & 0.03 & 3.15 & 2.45 & 3.74 & \dots \\ 3.45 & 0.41 & 0.46 & 1.60 & 4.80 & 0.25 & 0.89 & 5.52 & 0.63 & 4.71 & 3.47 & \dots \\ 7.06 & 0.72 & 4.79 & 0.58 & 0.31 & 8.81 & 0.32 & 0.16 & 0.57 & 0.37 & 2.78 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

modified K-means clustering

$$\mathbf{A} = \begin{bmatrix} 26.6 & 5.2 & 2.6 & \dots \\ 6.2 & 38.8 & 7.4 & \dots \\ 6.6 & 7.0 & 10.4 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Velo: **Ve**rify maximum link **lo**ads under failures and routing changes

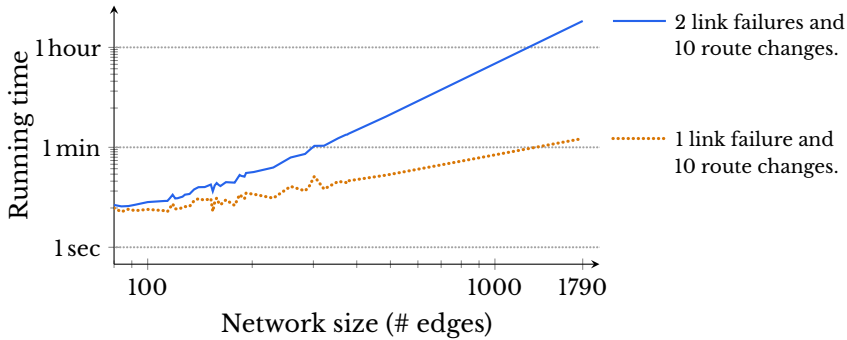
Search space reduction:

A single egress router maximizes link loads.

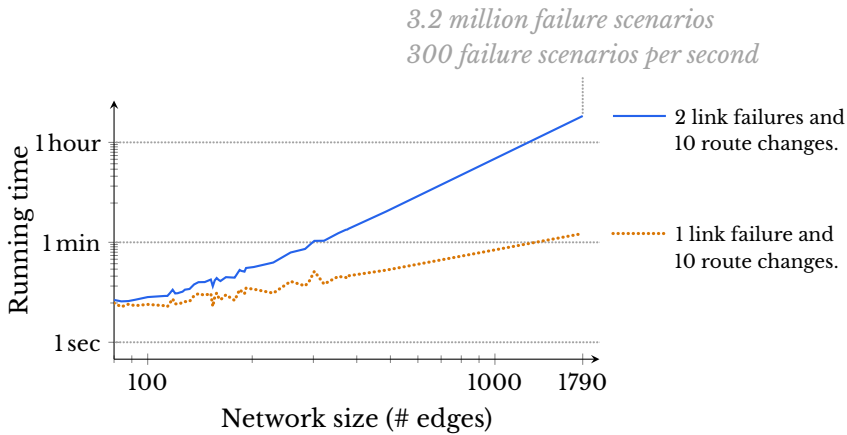
Input size reduction:

Cluster destination with similar traffic patterns.

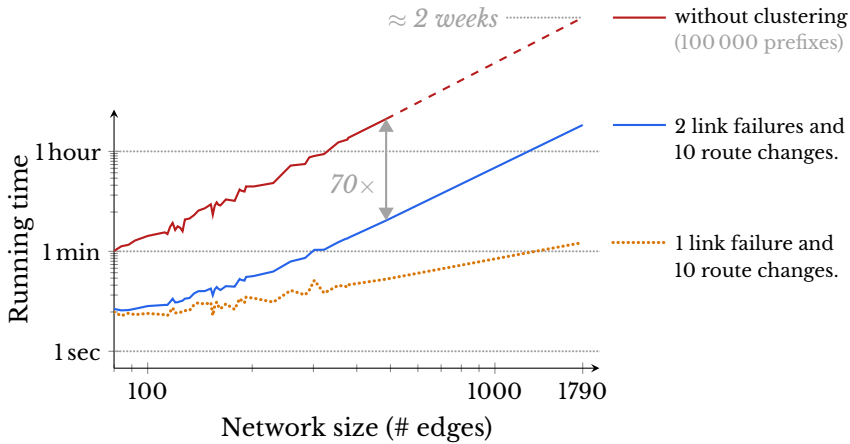
Velo finds the worst-case loads in *3 hours* for all 1790 links in an ISP.



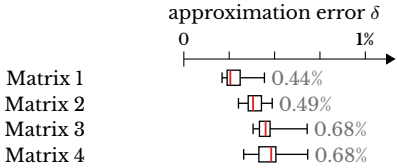
Velo finds the worst-case loads in *3 hours* for all 1790 links in an ISP.



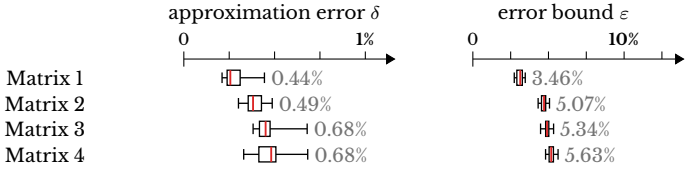
Velo finds the worst-case loads in *3 hours* for all 1790 links in an ISP.



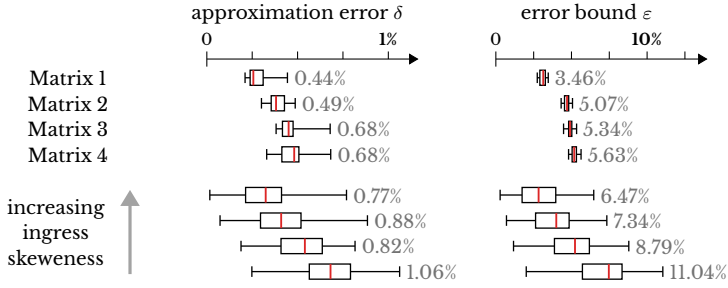
Velo achieves high accuracy with just 300 clusters.



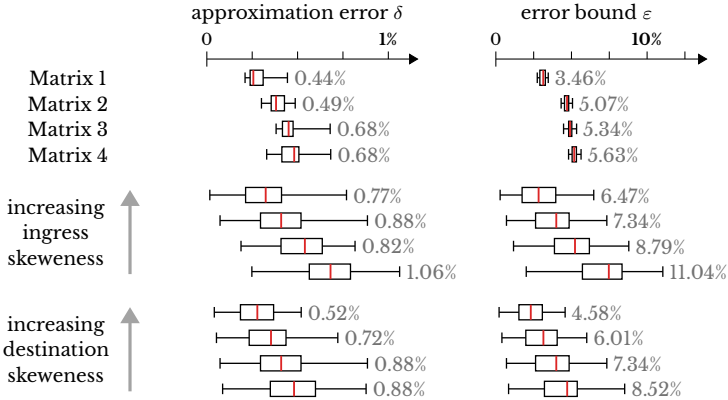
Velo achieves high accuracy with just 300 clusters.



Velo achieves high accuracy with just 300 clusters.



Velo achieves high accuracy with just 300 clusters.



Open problems and next steps.

Some destinations have more stable routes than others.

$\Rightarrow$ Can we use probabilistic techniques as well?

Open problems and next steps.

Some destinations have more stable routes than others.

⇒ Can we use probabilistic techniques as well?

Ingress traffic is not constant.

⇒ Can we reason about traffic shifts as well?

Velo: **Ve**rifying maximum link **lo**ads in a changing world

Given a ***network****,* ***configuration****, and a* ***traffic matrix****,*
Velo *finds the* ***worst-case load*** *of* ***all links***
under ***routing changes and failures****.*

Search space reduction:

A single egress router maximizes link loads.

Input size reduction:

Cluster destination with similar traffic patterns.