Transient Forwarding Anomalies and How to Find Them

...

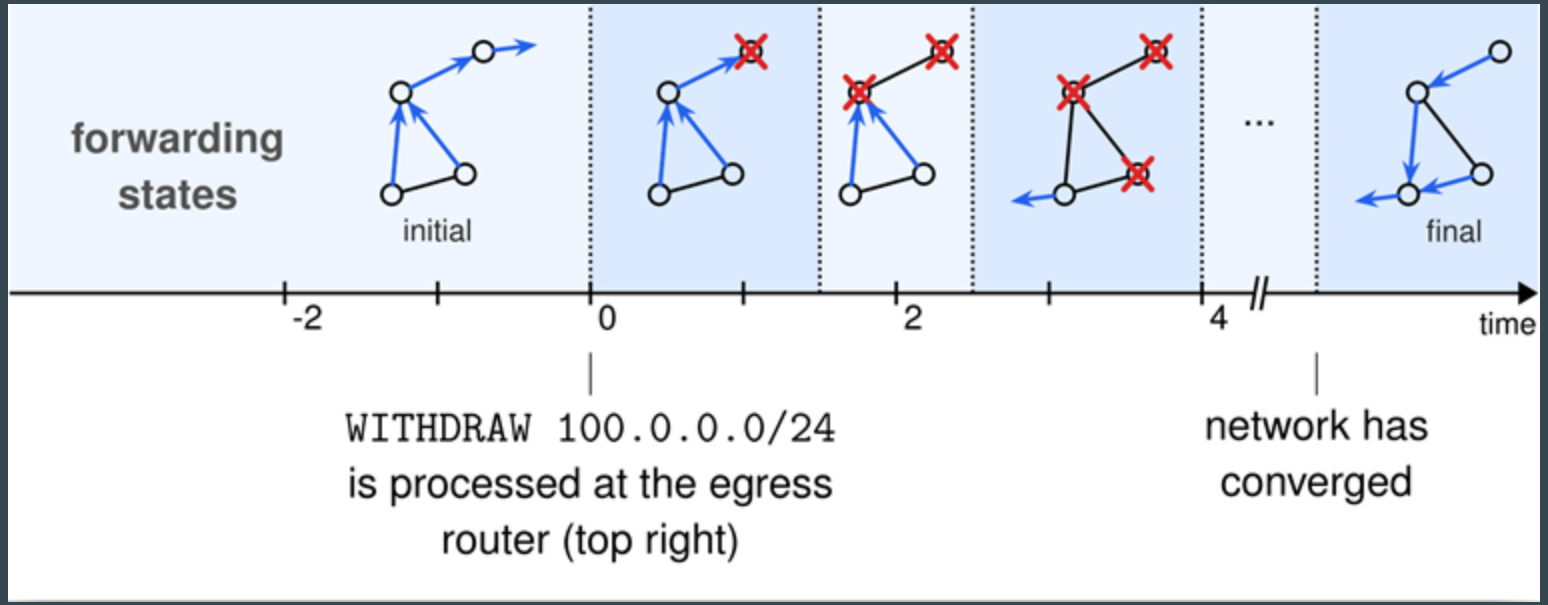
Roland Schmid, Tibor Schneider, Georgia Fragkouli, Laurent Vanbever

CoNEXT 2025
2nd of December, 2025

Networked Systems
ETH Zürich — seit 2015

ETH zürich

Problem: Analyzing disruptions **during routing convergence.**



Analyzing transient network disruptions is **hard**.

Analyzing transient network disruptions is **hard**.

Network disruptions ...

... depend on data-plane state across time,

Analyzing transient network disruptions is **hard**.

Network disruptions ...

... depend on data-plane state across time,

... and may differ across prefixes.

Analyzing transient network disruptions is **hard**.

Network disruptions ...

... depend on data-plane state across time,

... and may differ across prefixes.

→ Naively probing the full data-plane state continuously overwhelms networks.

Previous work is either **inaccurate** or has **low prefix coverage**.

IEEE INFOCOM 2005

Happy Packets to You!

Randy Bush, Timothy G. Griffin, Jun Li, Zhuoqing M. Mao, Eric Purpus, Daniel Stutzbach

Paper ID: 1568938735
Number of pages: 12

***Abstract*— Previous studies of the Internet control plane examined its quality from a number of different points of view, but very few considered the performance of the data plane as a measure of control plane performance. For example, a large number of routing updates during some period or slow routing convergence, are often cited as an indication of a poorly designed or behaving routing protocol. However, if the customers' packets are being delivered well, which we term "happy packets", while routers can handle routing updates without being overloaded, then how significant are these control plane measurements?**

We believe packet happiness should be the primary metric for measuring control plane performance, especially the quality of routing. Since handling routing changes is the fundamental function of any routing protocol, in this paper we measure packet happiness during controlled routing changes. We obtain results by using established data plane metrics – delay, drop, jitter and reordering (DDJ&R) – in order to evaluate BGP, the *de facto* standard inter-domain routing protocol. Moreover, we compare the results with routing updates observed by RouteViews, a commonly used resource for studying the control plane. We establish that generally there is no clear correlation between the number and duration of routing updates derived from partial views and the performance of the data plane, further cautioning the usage of metrics from the control plane alone in evaluating control plane quality.

***Index Terms*— control plane, data plane, BGP, happy packet, network measurement**

I. Introduction

We frequently hear comments about Internet control plane quality, such as

- Internet routing is fragile and collapsing,
- Yesterday was a bad routing day,
- BGP is broken,

To evaluate Internet control plane quality, a variety of measures have been used, including number and frequency of routing updates [1], [2], number of prefixes [3], size of routing tables [4], and the speed or completeness of routing convergence [5], [6]. For example, significant numbers of BGP updates are often equated with Internet instability [1], [2]. One such comment that is often expressed by people studying BGP is "There are too many BGP updates, so BGP must be broken."

But what is good routing? How can one decide that a particular measurement shows one routing scheme is superior to another? What metrics should be used?

While people often contend that one measurement is better or worse than another when measuring control plane quality, the measure which ultimately counts is whether the customer's packets reach their intended destination with good performance. If the customers' packets are happy, the routing system, and other components in the control plane, are doing their job well. So, while all those metrics above are indeed important, packet performance should be the primary metric for evaluating control plane performance.

It is indeed true that sometimes Internet routing may experience huge routing updates, but perhaps they are just like white blood cells; although their presence may signal a problem, they are often part of the cure, not necessarily part of the problem. As long as routers will not fall over due to a sharp load increase, it is not clear the control plane is in danger. Furthermore,

Computer Communications 129 (2018) 166–183

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A fine-grained multi-source measurement platform correlating routing transitions with packet losses

Pascal Merindol[a,*], Pierre David[a], Jean-Jacques Pansiot[a], Francois Clad[b], Stefano Vissicchio[c]

[a] Université de Strasbourg, France
[b] Cisco Systems, USA
[c] University College London, UK

ARTICLE INFO

Keywords:
Active and passive measurements
IP routing convergence
Control plane
Date plane
Management plane

ABSTRACT

In this paper, we are interested in the relationship between packet losses and routing changes in an operational network. To do so we designed and deployed DCART, a monitoring platform over RENATER, the French research and education network. Our platform collects four data sources using both active and passive measurements in order to unveil their temporal correlations. Active probing allows especially for measuring packet losses on specifically crafted data flows. Those flows explore several load balanced paths and ease the revelation of forwarding loops. Passive monitoring is achieved by listening to all routing updates from IS-IS, the intra-domain routing protocol in use, and by retrieving tickets generated by the Network Operations Center (NOC).

During our monitoring campaign, we observe that most of the series of loss were correlated to routing events either because routing changes lead to inconsistent state transitions, or because faulty – and so lossy – links trigger numerous periods of link flapping. In particular, we show that losses due to forwarding loops resulting from inconsistent routing states are quite common when links come back after an outage. We also show that link flapping sometimes induce very long lasting lossy periods frequently unnoticed by the NOC. A lightweight monitoring platform such as DCART could be used to better anticipate recurrent network outages and to improve the ticketing system.

1. Introduction

Internet service providers (ISP) seek a very high degree of reliability for their network, the 99.999% (so called five nines) availability ratio is often mentioned. This demand is triggered both by applications requiring strong network quality such as voice calls, and by clients asking Service Level Agreements (SLA) that require to be verified

education network); it gathers data from several sources. Constant rate data probes monitor the data plane, recording losses, ICMP error messages, path changes and one-way delay variations. Carefully crafted one-way probes allow an ISP to detect and locate forwarding loops and to monitor different ECMP[1] paths. An Interior Gateway Protocol (IGP) listener, in our case IS-IS [1], collects all changes in the intra-domain routing control plane. The management plane is tracked through tickets issued by the Network Operations Center (NOC). A design goal of this platform is to be as independent as possible from the network infrastructure, so data probers as well as the IGP listener are implemented on commodity devices, using only open source software and

Goals

We want to measure transient network disruptions:

Goals

We want to measure transient network disruptions:

- accurately,

Goals

We want to measure transient network disruptions:

- accurately,
- for all prefixes,

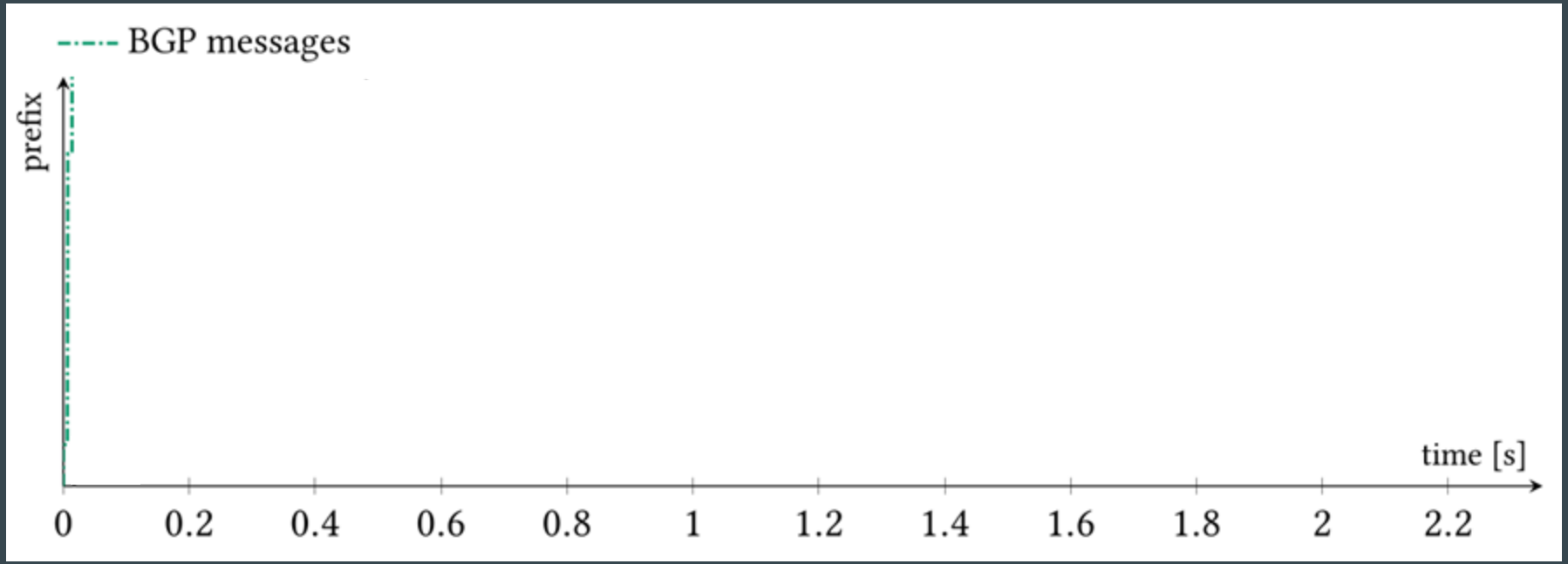
Goals

We want to measure transient network disruptions:

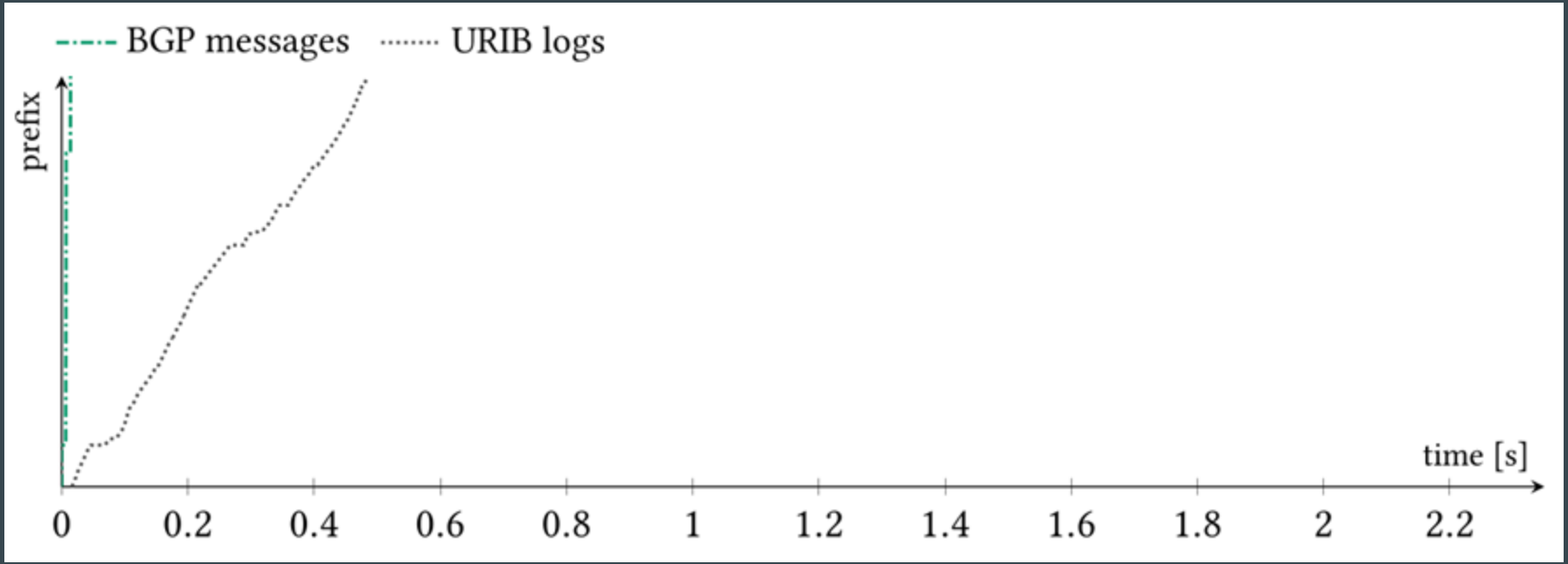
- accurately,
- for all prefixes,
- without impairing the network‘s performance.

Idea: Using **router-local information** to infer disruptions.

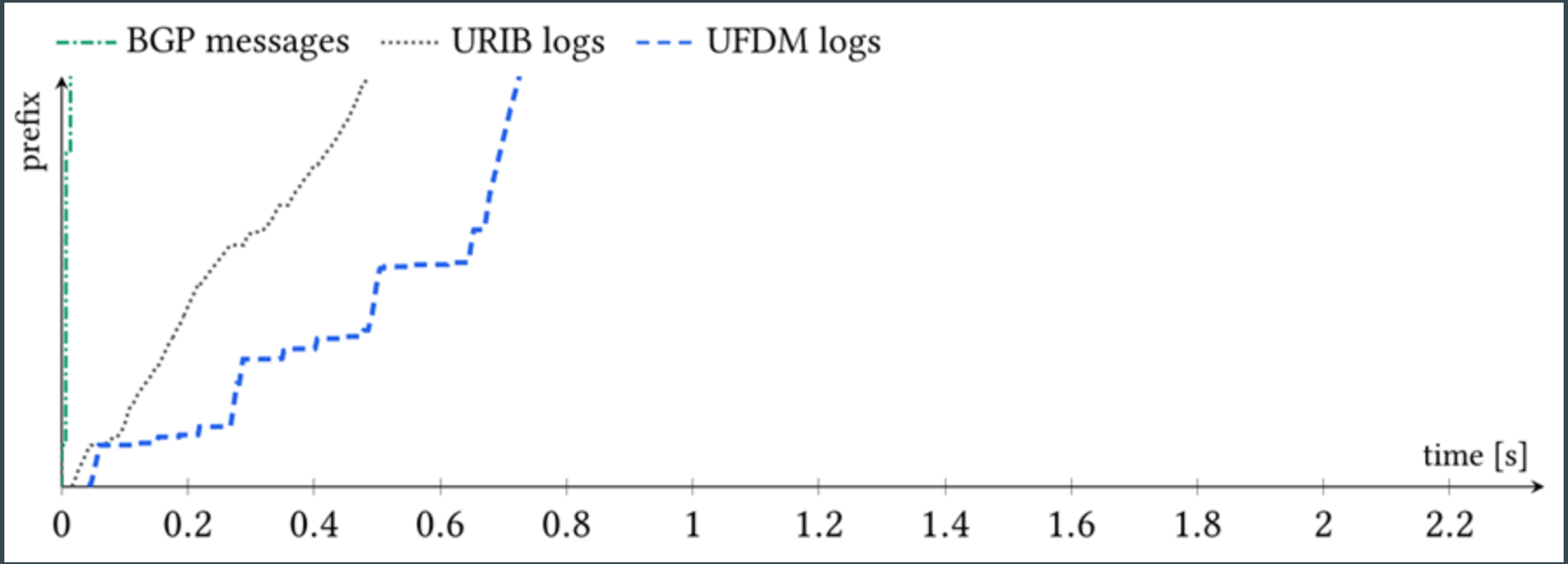
Idea: Using **router-local information** to infer disruptions.



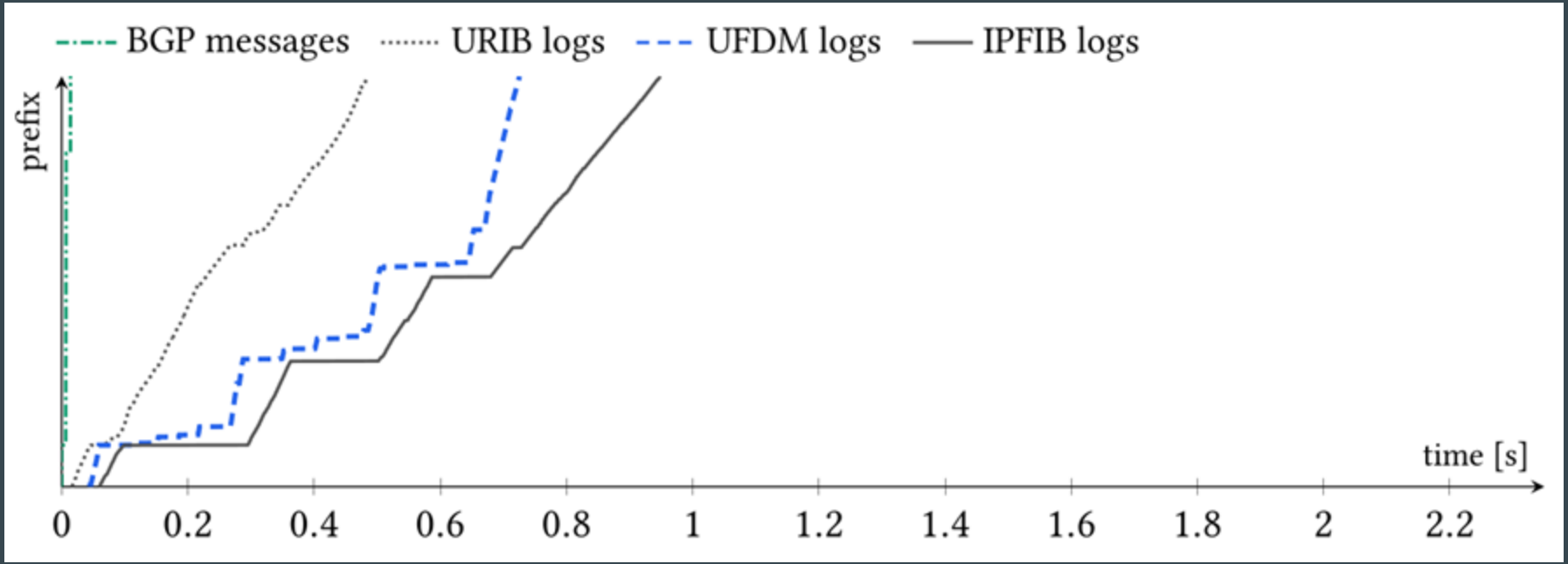
Idea: Using **router-local information** to infer disruptions.



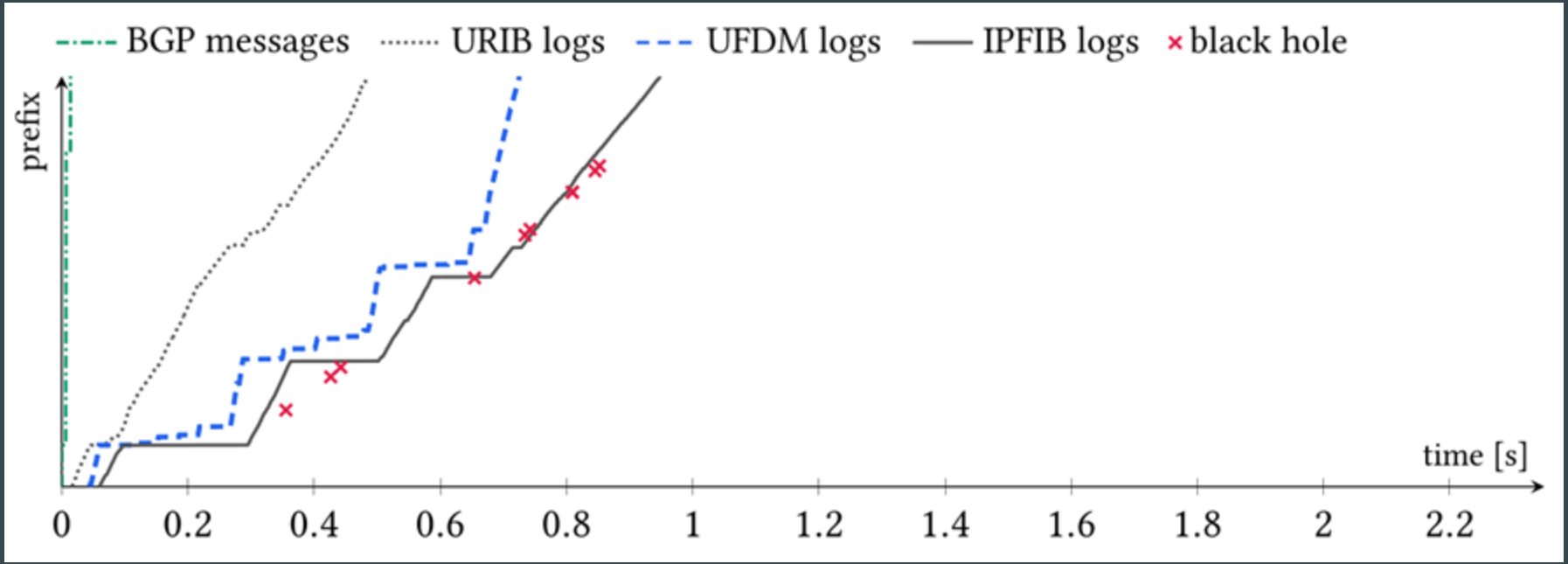
Idea: Using **router-local information** to infer disruptions.



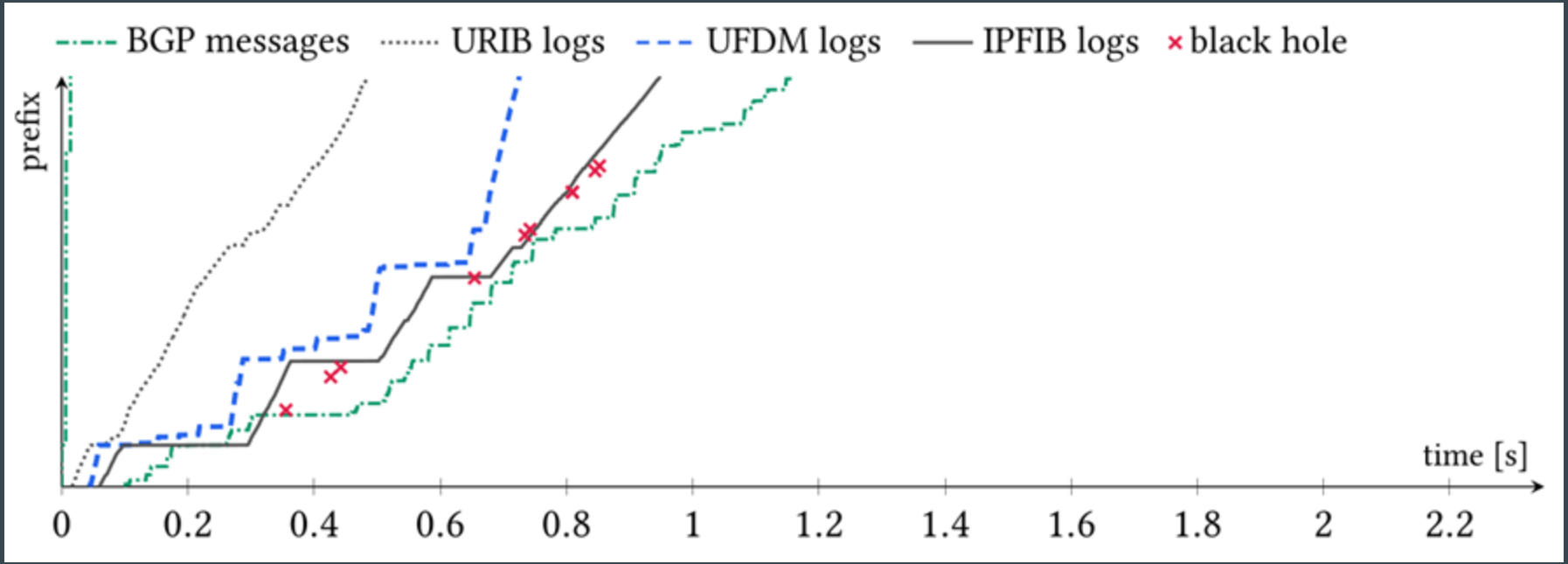
Idea: Using **router-local information** to infer disruptions.



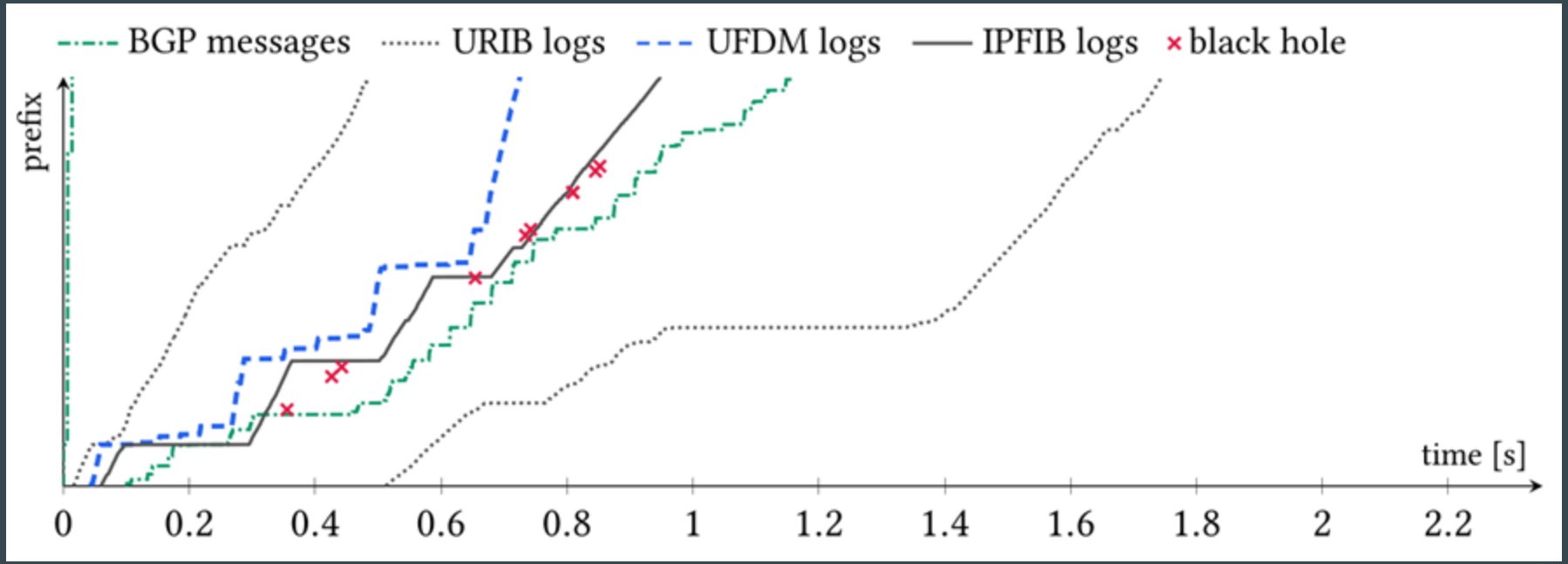
Idea: Using **router-local information** to infer disruptions.



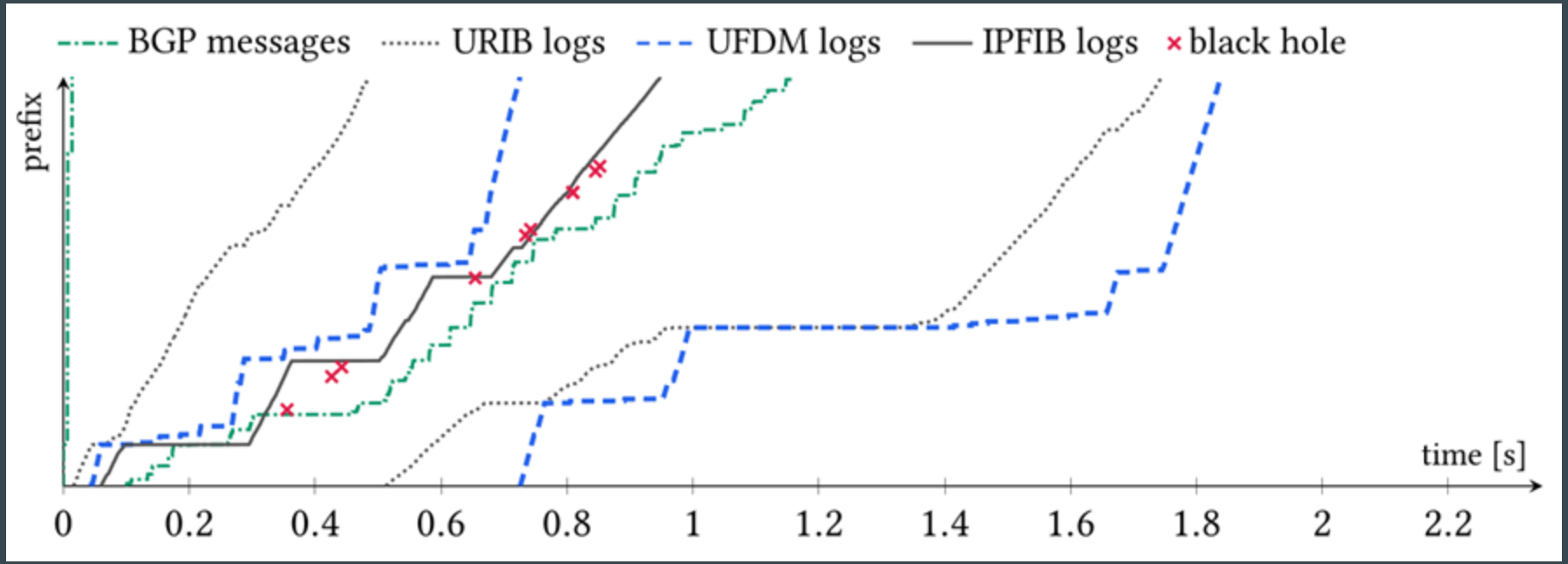
Idea: Using **router-local information** to infer disruptions.



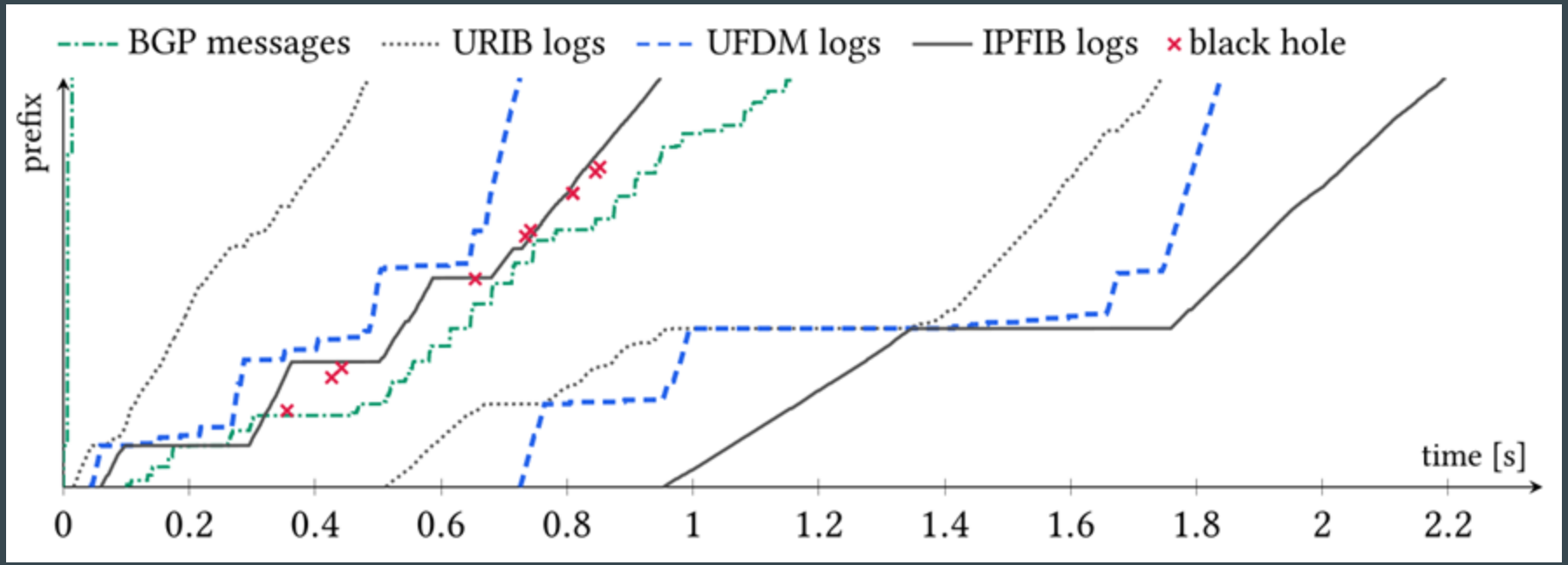
Idea: Using **router-local information** to infer disruptions.



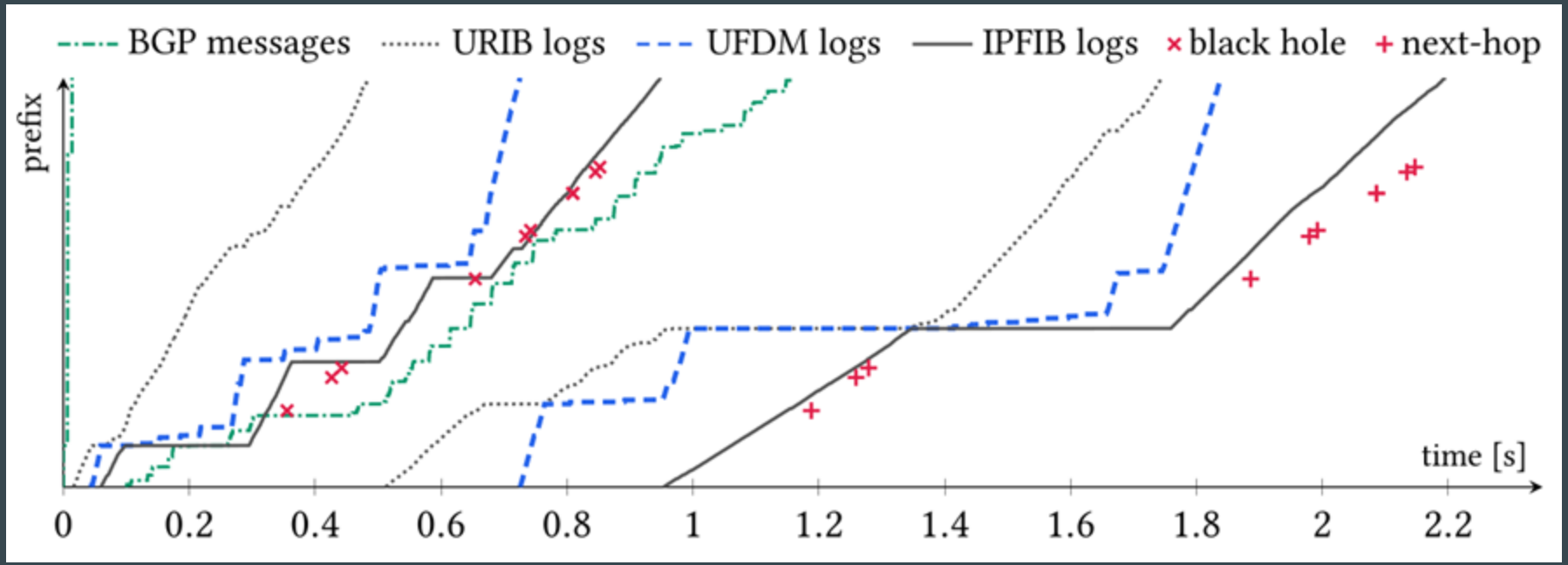
Idea: Using **router-local information** to infer disruptions.



Idea: Using **router-local information** to infer disruptions.



Idea: Using **router-local information** to infer disruptions.

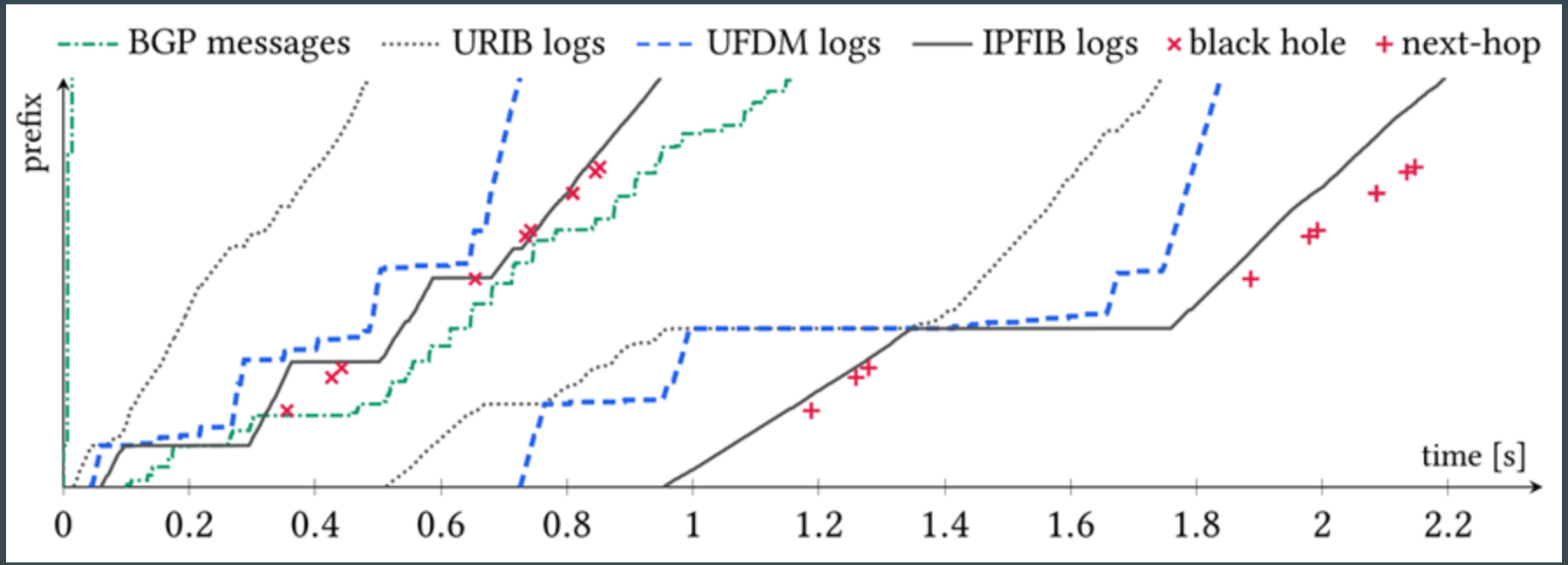


TRIX: Bridging the gap from control- to data-plane information.

Challenges:

1. Router-local information is **incomplete**.

Insight #1: Control-plane **simulation** to infer data-plane updates.



TRIX: Bridging the gap from control- to data-plane information.

Challenges:

1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

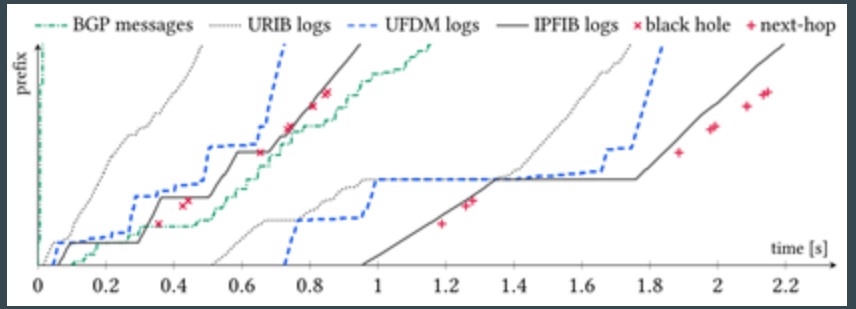
TRIX: Bridging the gap from control- to data-plane information.

Challenges:

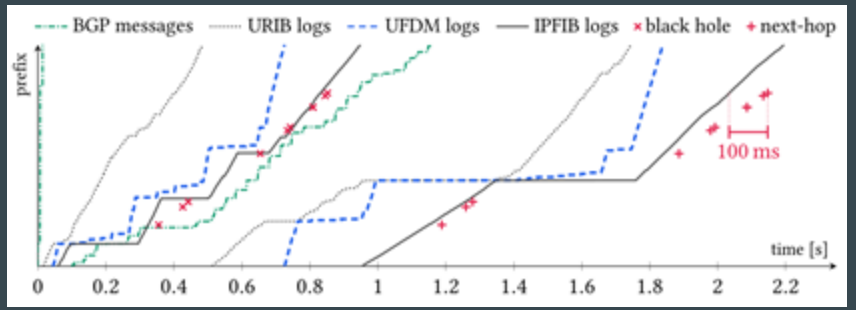
1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

2. Router-local information is **inaccurate**.

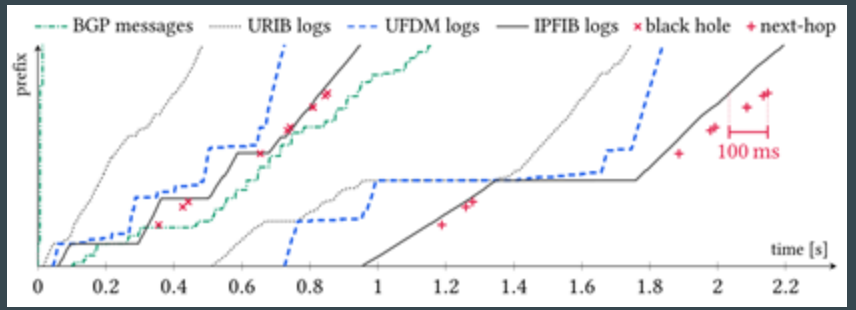
Insight #2: Modeling the hardware **bottleneck** at the FIB.



Insight #2: Modeling the hardware **bottleneck** at the FIB.



Insight #2: Modeling the hardware **bottleneck** at the FIB.



- We model the **hardware bottleneck** at the FIB with a simple queuing model:

$$up_i = \max\left(t_i + T_{\text{delay}}\,,\, up_{i-1}\right) + \begin{cases} T_{\text{drop}} & \text{if } nh_i = \emptyset \\ T_{\text{fw}} & \text{otherwise} \end{cases}$$

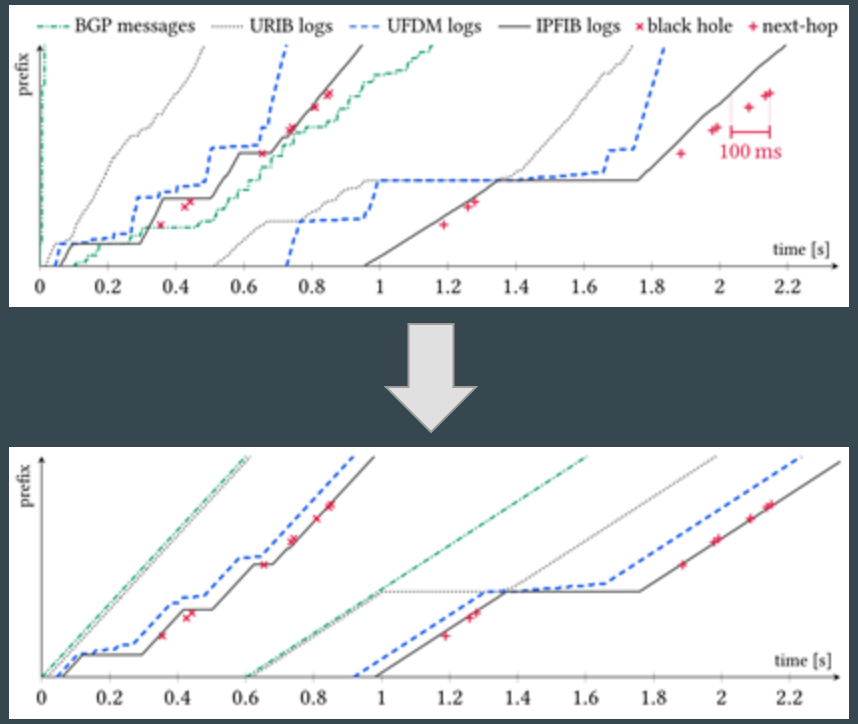
- For illustration purposes, we set T_delay = 0.

Insight #2: Modeling the hardware **bottleneck** at the FIB.

- We model the **hardware bottleneck** at the FIB with a simple queuing model:

$$up_i = \max\left(t_i + T_{\text{delay}}\,,\, up_{i-1}\right) + \begin{cases} T_{\text{drop}} & \text{if } nh_i = \emptyset \\ T_{\text{fw}} & \text{otherwise} \end{cases}$$

- For illustration purposes, we set T_delay = 0.



TRIX: Bridging the gap from control- to data-plane information.

Challenges:

1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

2. Router-local information is **inaccurate**.
   → Modeling the **hardware bottleneck** at the FIB.

TRIX: Bridging the gap from control- to data-plane information.

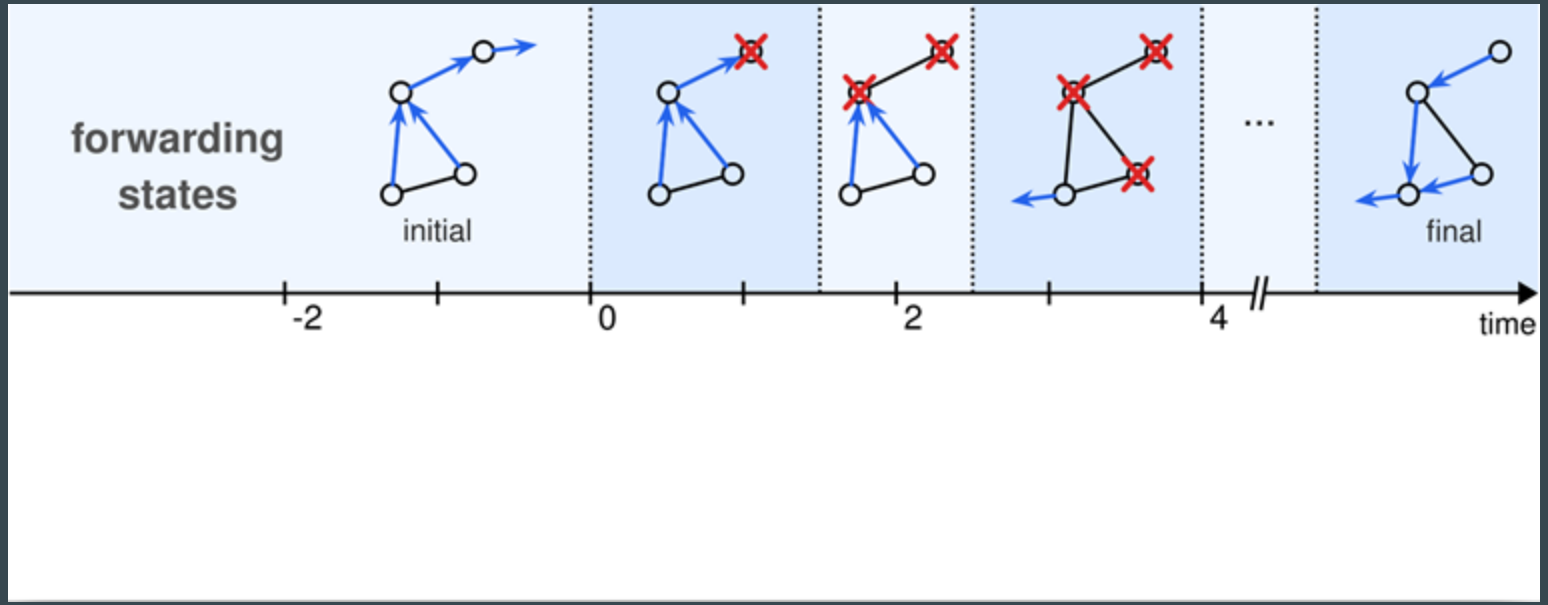
Challenges:

1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

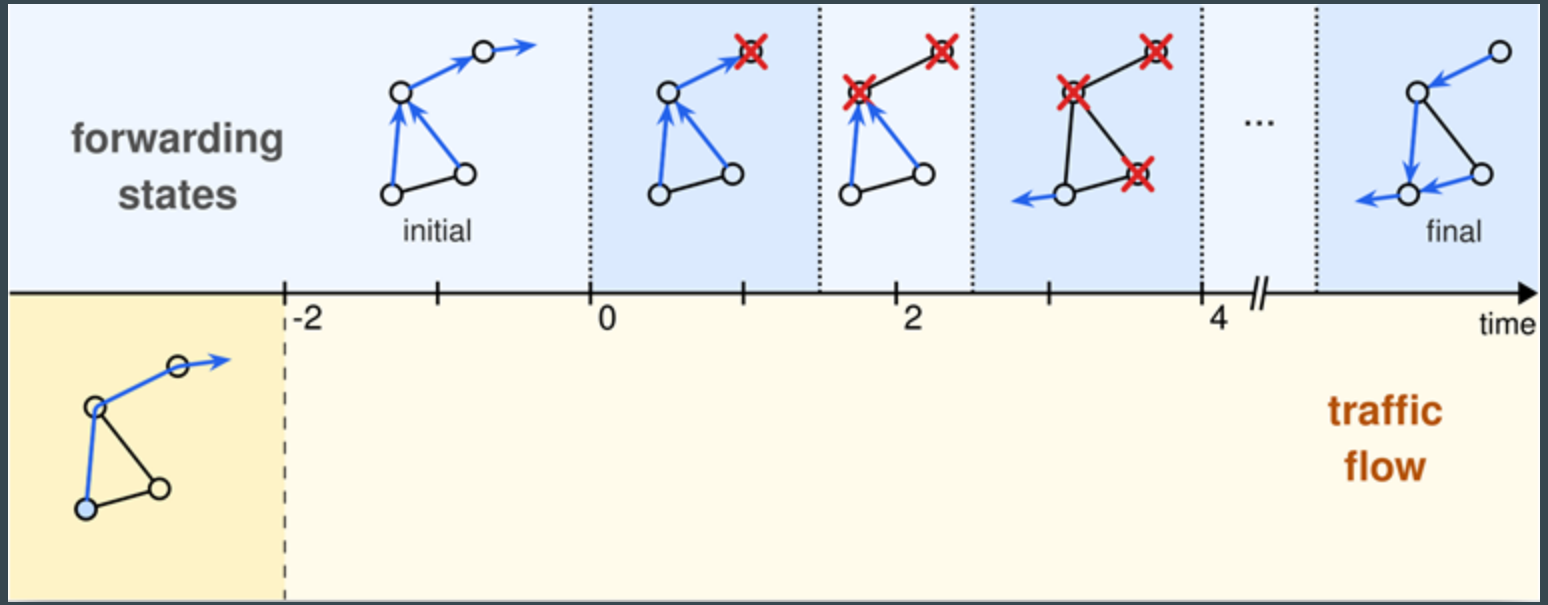
2. Router-local information is **inaccurate**.
   → Modeling the **hardware bottleneck** at the FIB.

3. Reconstructing **network-wide disruptions** from router-local information.

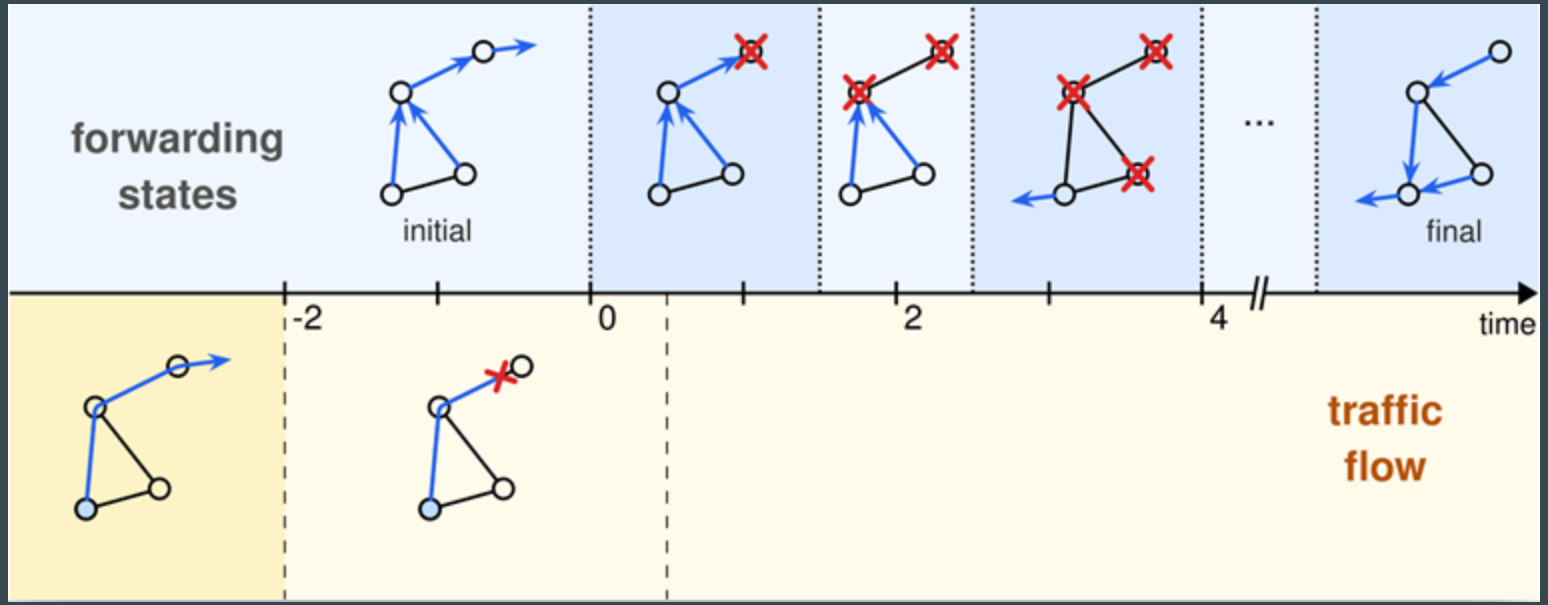
Insight #3: Modeling traffic propagation through the network.



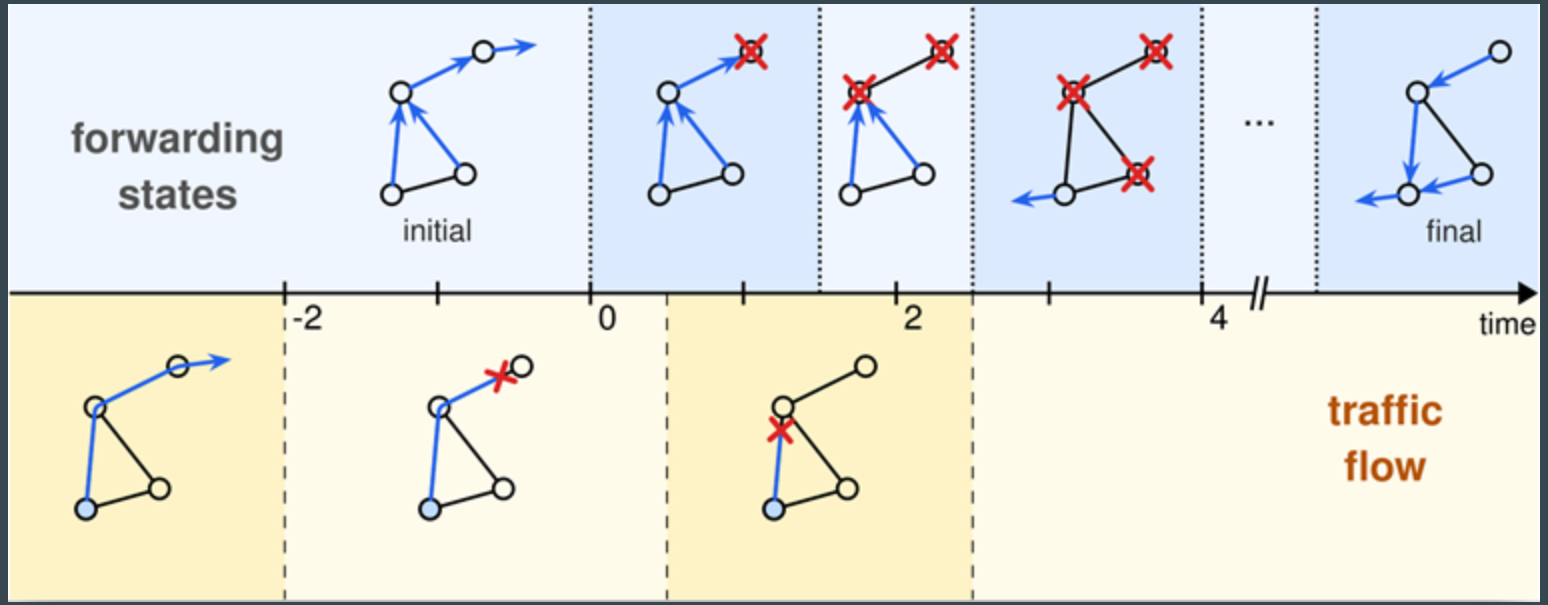
Insight #3: Modeling traffic propagation through the network.



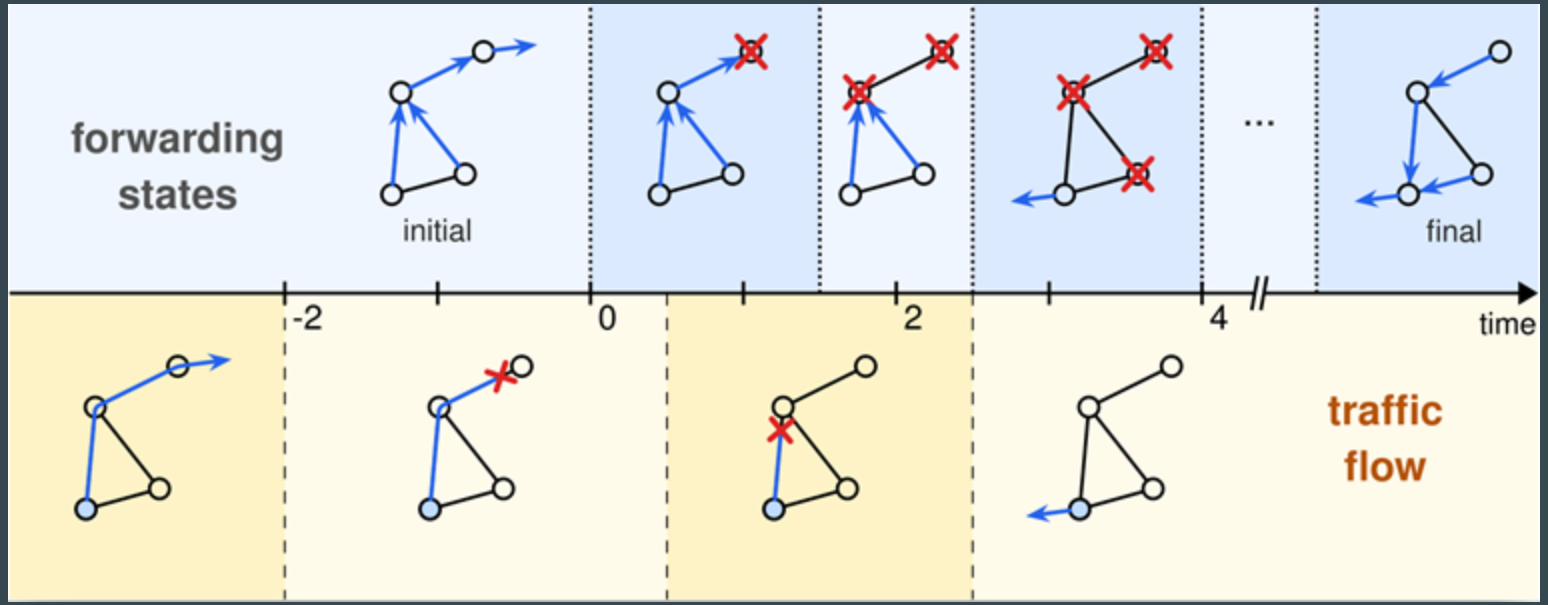
Insight #3: Modeling traffic propagation through the network.



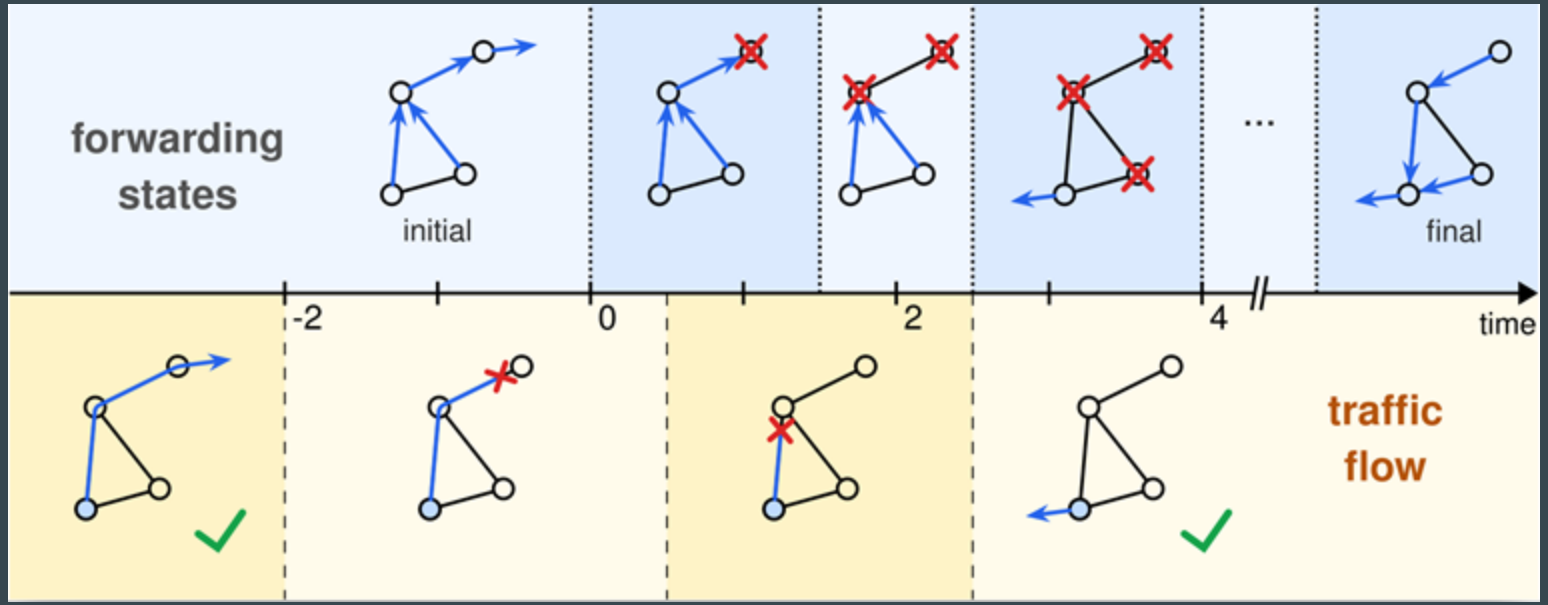
Insight #3: Modeling traffic propagation through the network.



Insight #3: Modeling traffic propagation through the network.



Insight #3: Modeling traffic propagation through the network.



TRIX: Bridging the gap from control- to data-plane information.

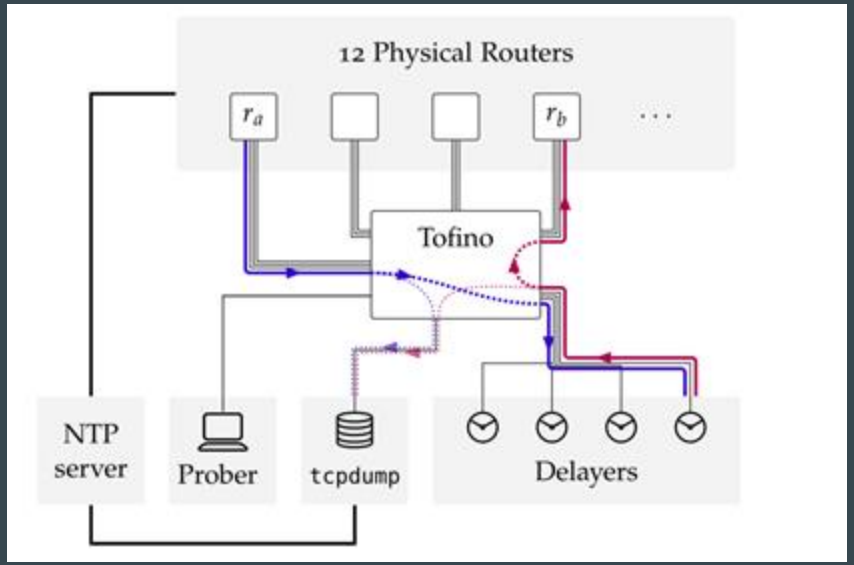
Challenges:

1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

2. Router-local information is **inaccurate**.
   → Modeling the **hardware bottleneck** at the FIB.

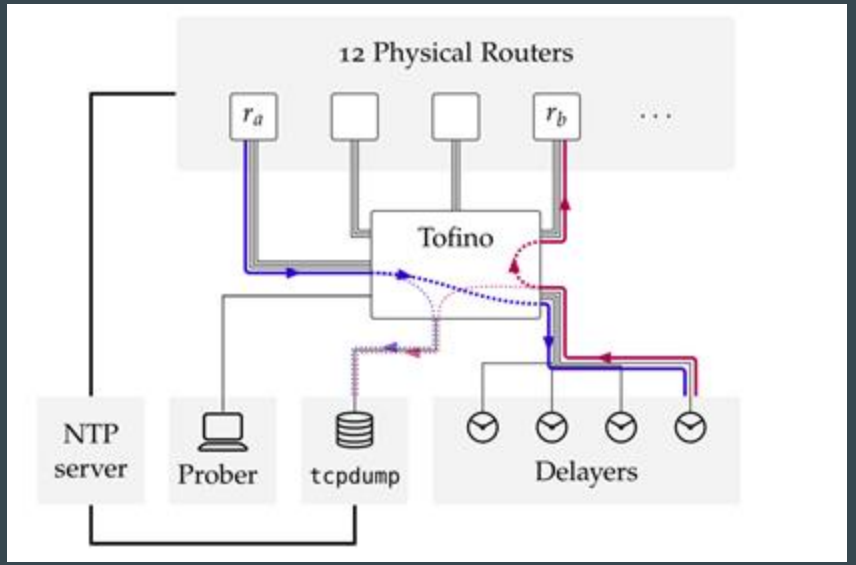
3. Reconstructing **network-wide disruptions** from router-local information.
   → Modeling **traffic propagation** through the network.

Evaluation


12 Physical Routers
r_a
r_b
...
Tofino
NTP server
Prober
tcpdump
Delayers

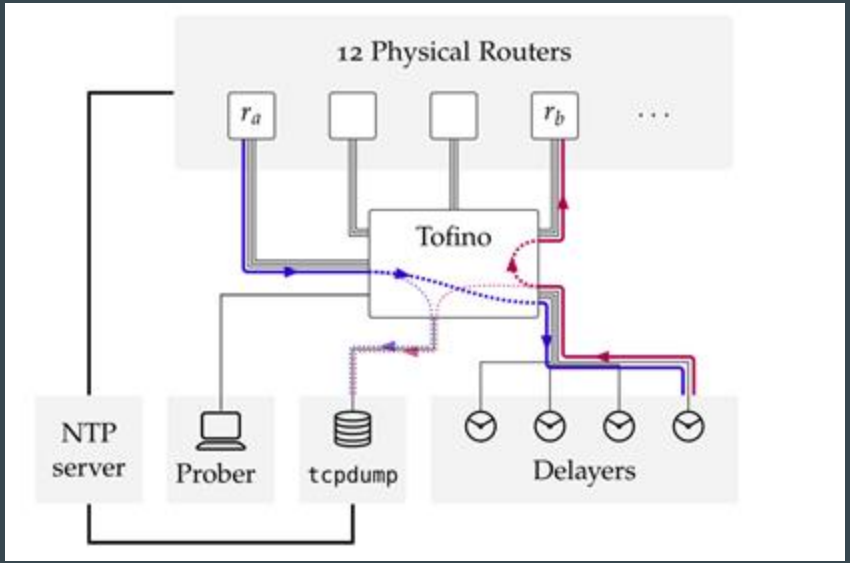
Evaluation

- Testbed with **12 physical hardware routers** emulating the Abilene network topology.



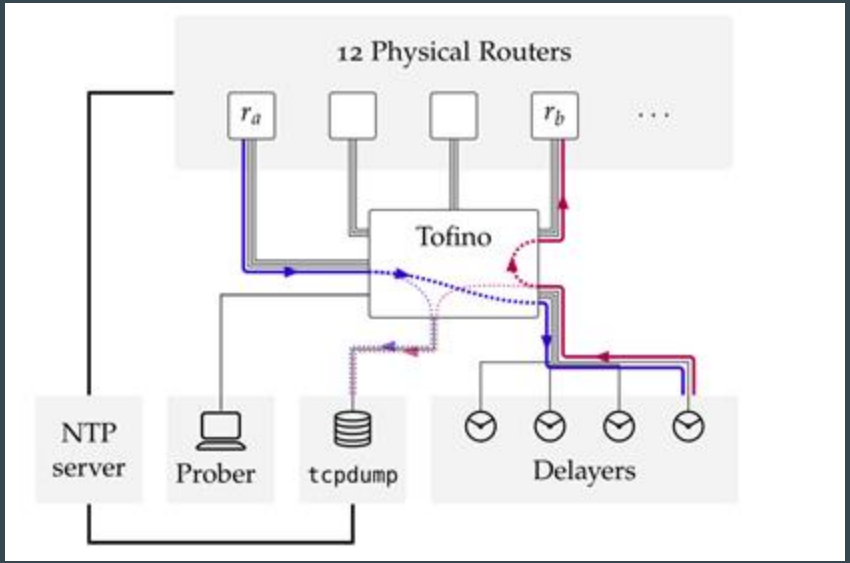
Evaluation

- Testbed with **12 physical hardware routers** emulating the Abilene network topology.
- Collect **synchronized** router logs and BGP messages to infer transient behavior.

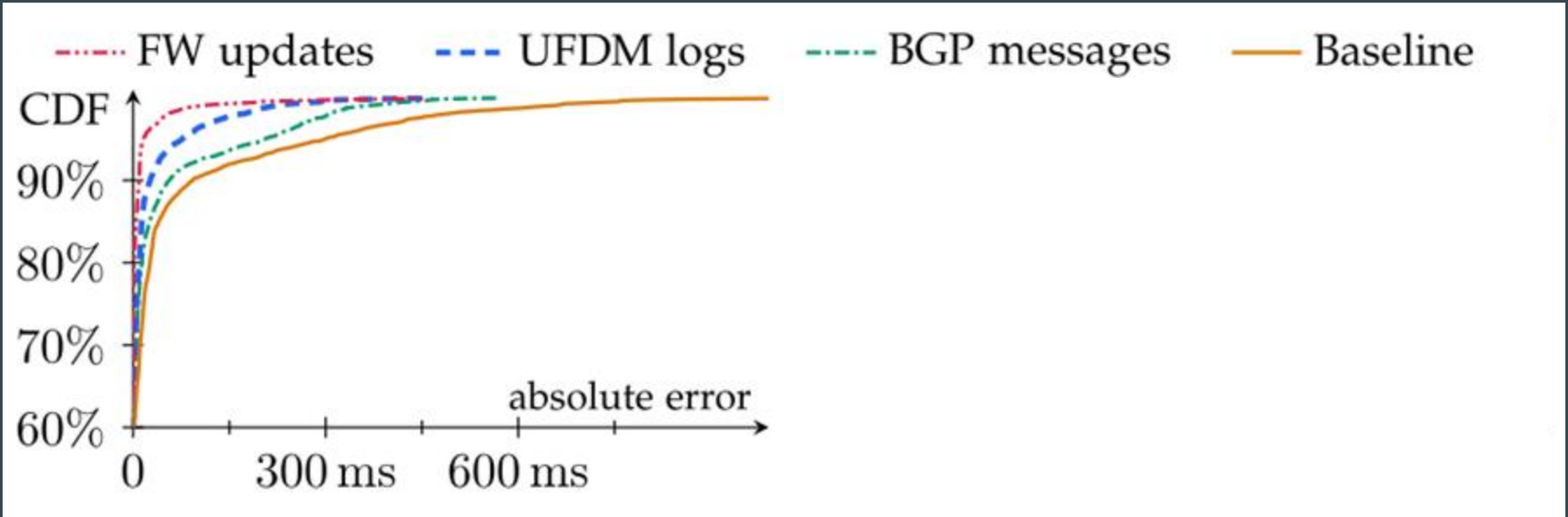


Evaluation

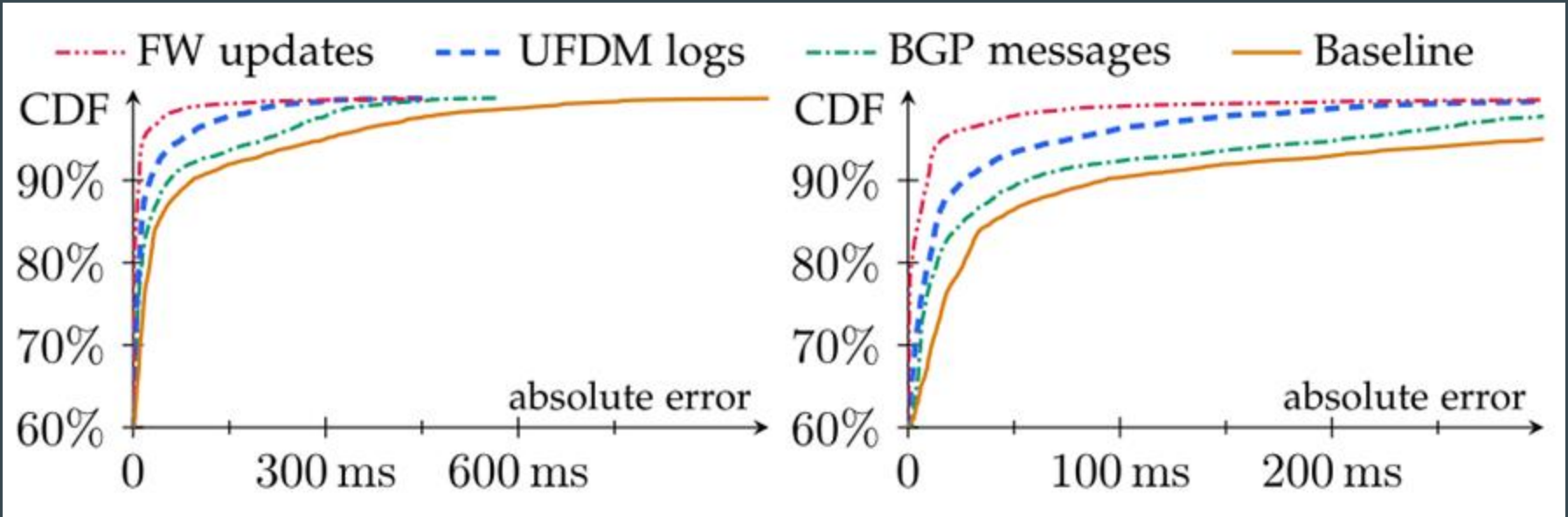
- Testbed with **12 physical hardware routers** emulating the Abilene network topology.
- Collect **synchronized** router logs and BGP messages to infer transient behavior.
- Gather ground-truth violation times from actively **probing 10 prefixes**.



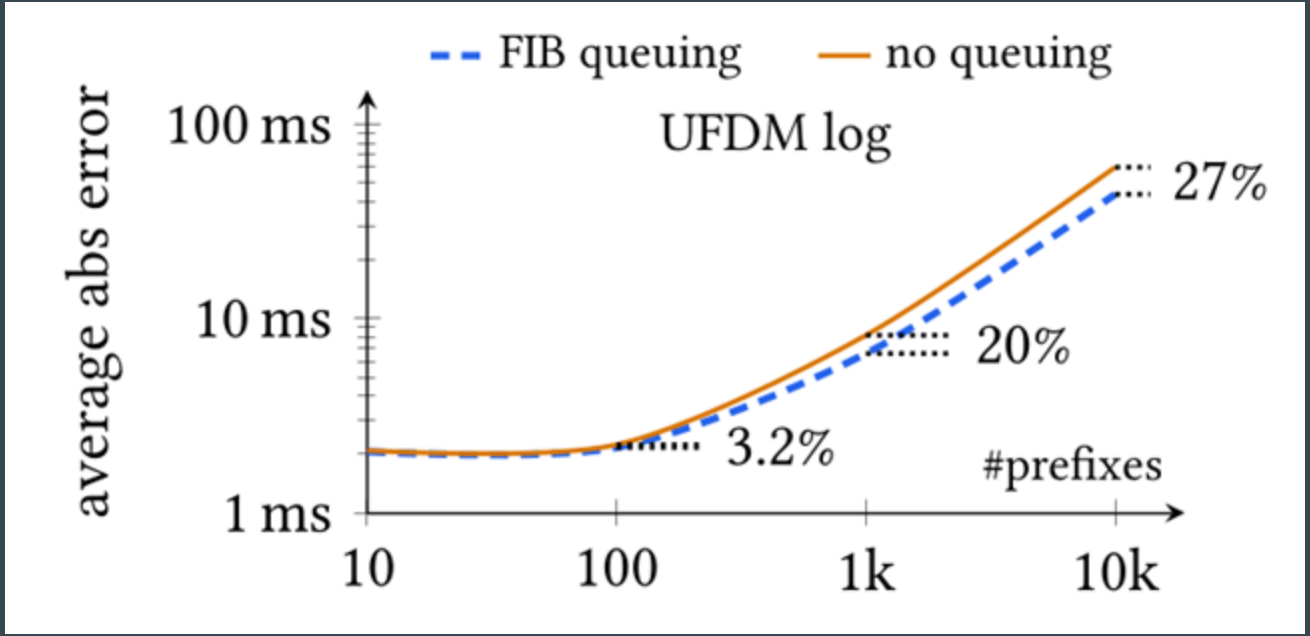
Overall, TRIX infers violations up to 13-25ms accurate on average.



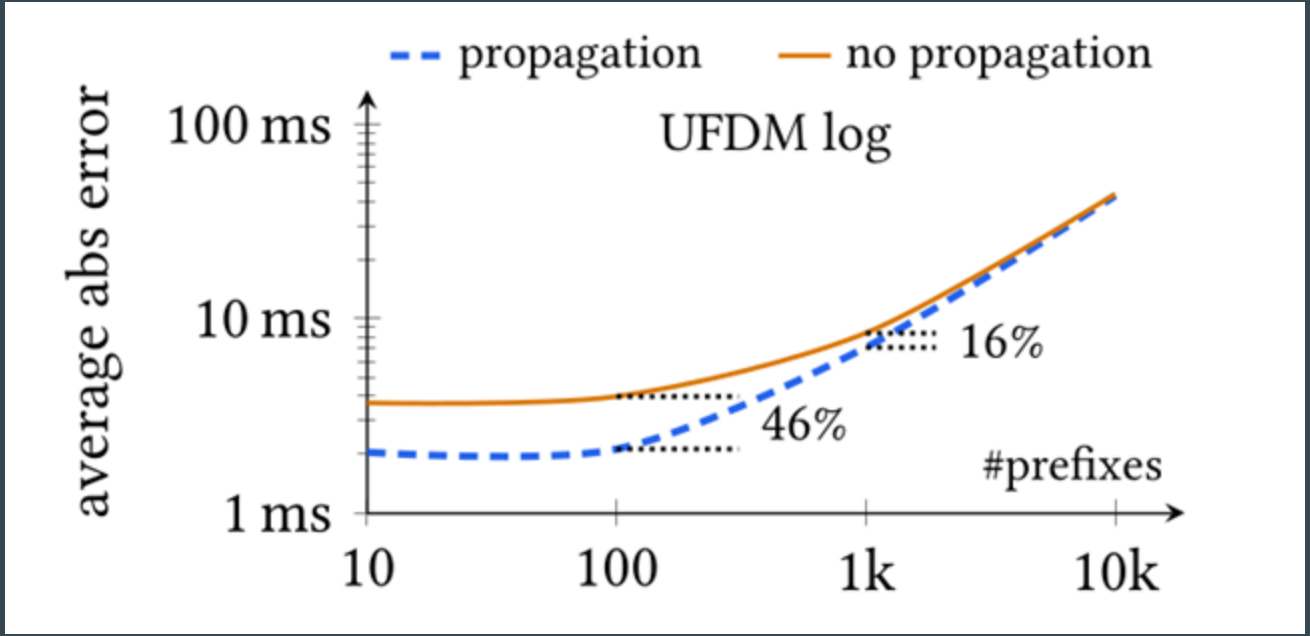
Overall, TRIX infers violations up to 13-25ms accurate on average.



The FIB queuing model boosts TRIX' accuracy by 25.1% on average.



Modeling propagation improves TRIX' accuracy by up to 50%.



Recap: TRIX uses **router-local** information to infer disruptions.

Recap: TRIX uses **router-local** information to infer disruptions.

Key insights:

1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

Recap: TRIX uses **router-local** information to infer disruptions.

Key insights:

1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

2. Router-local information is **inaccurate**.
   → Modeling the **hardware bottleneck** at the FIB.

Recap: TRIX uses **router-local** information to infer disruptions.

Key insights:

1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

2. Router-local information is **inaccurate**.
   → Modeling the **hardware bottleneck** at the FIB.

3. Reconstructing **network-wide disruptions** from router-local information.
   → Modeling **traffic propagation** through the network.

Recap: TRIX uses **router-local** information to infer disruptions.

Key insights:

1. Router-local information is **incomplete**.
   → Control-plane **simulation** to infer data-plane updates.

2. Router-local information is **inaccurate**.
   → Modeling the **hardware bottleneck** at the FIB.

3. Reconstructing **network-wide disruptions** from router-local information.
   → Modeling **traffic propagation** through the network.

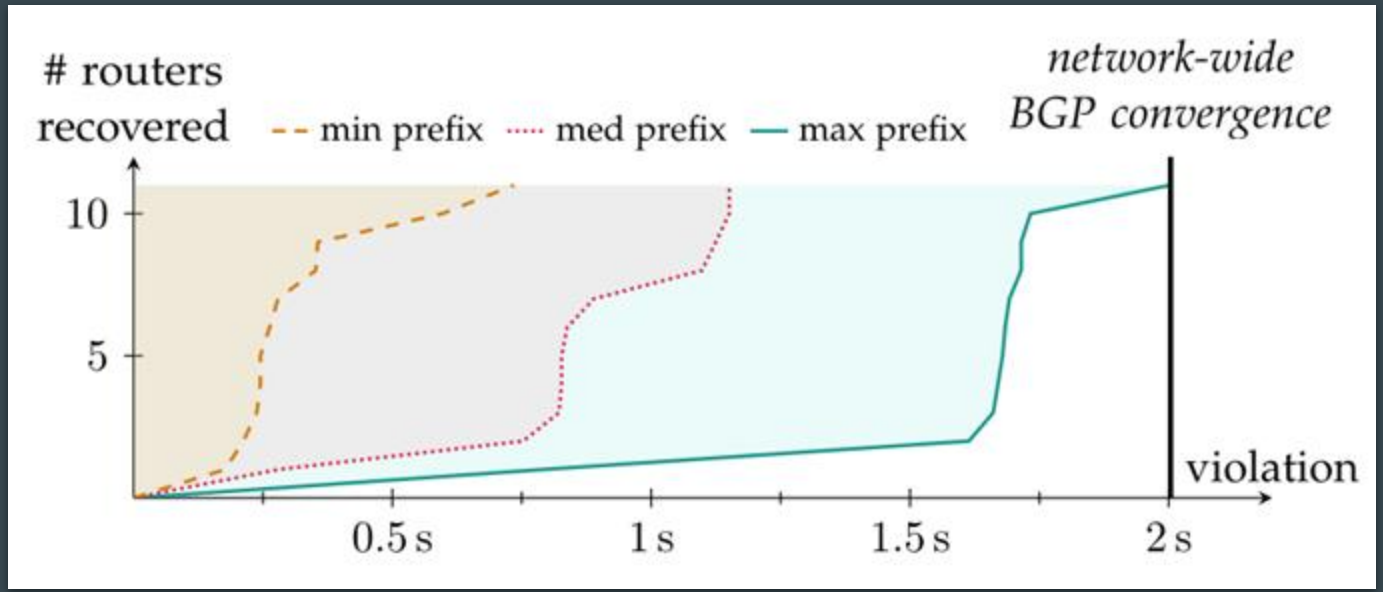
Paper Link

Networked Systems
ETH Zürich — seit 2015

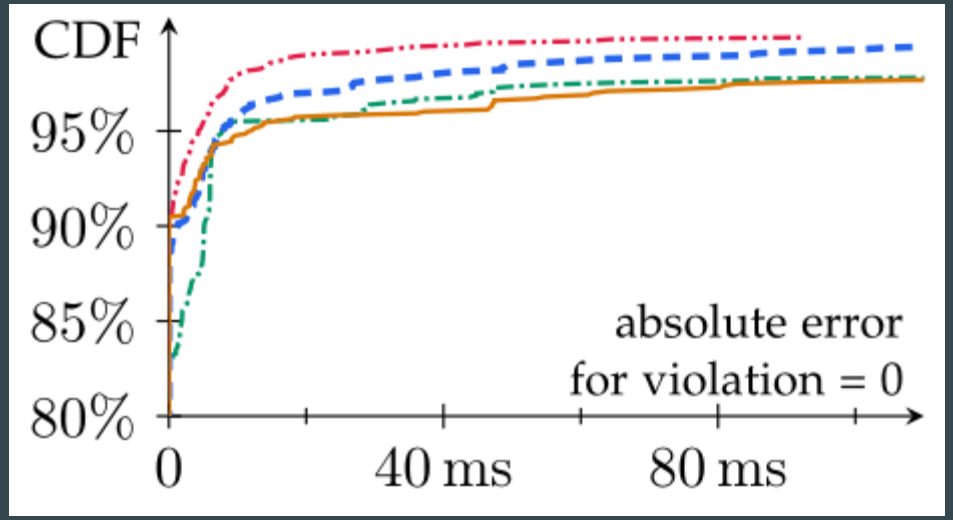
ETH zürich

Additional Material

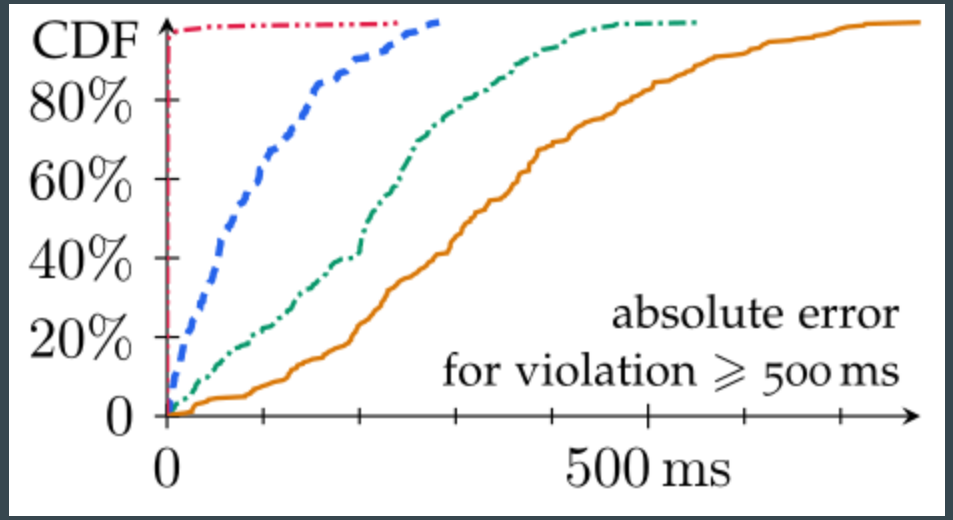
Measuring the BGP convergence time is not enough.



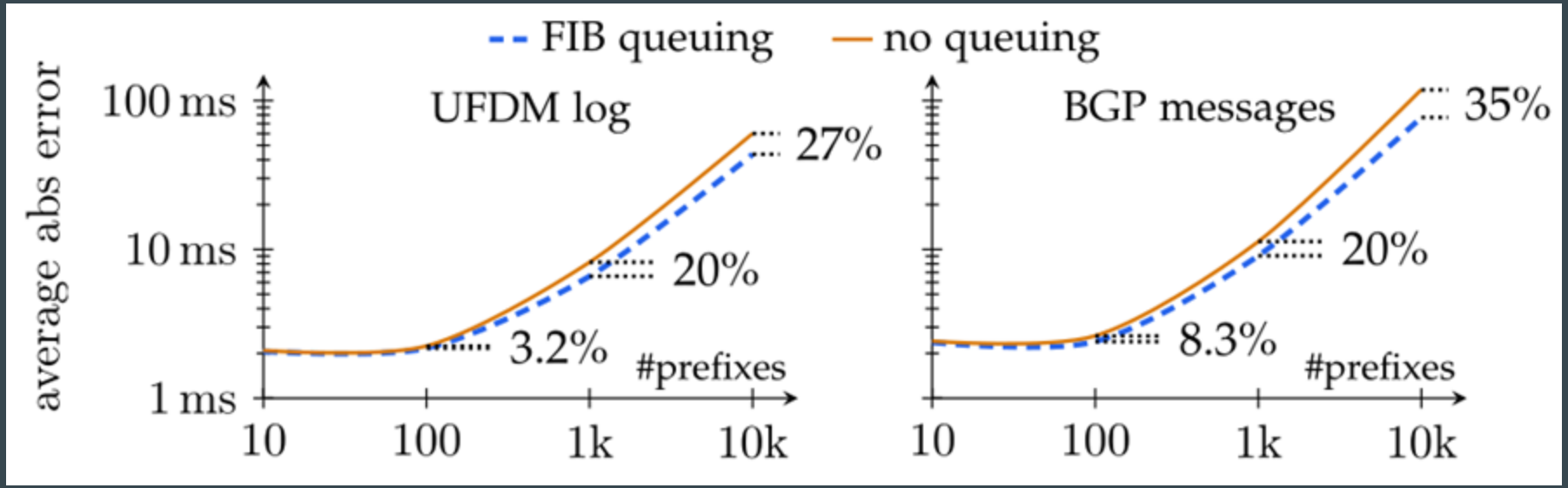
TRIX mostly detects when there is no violation.

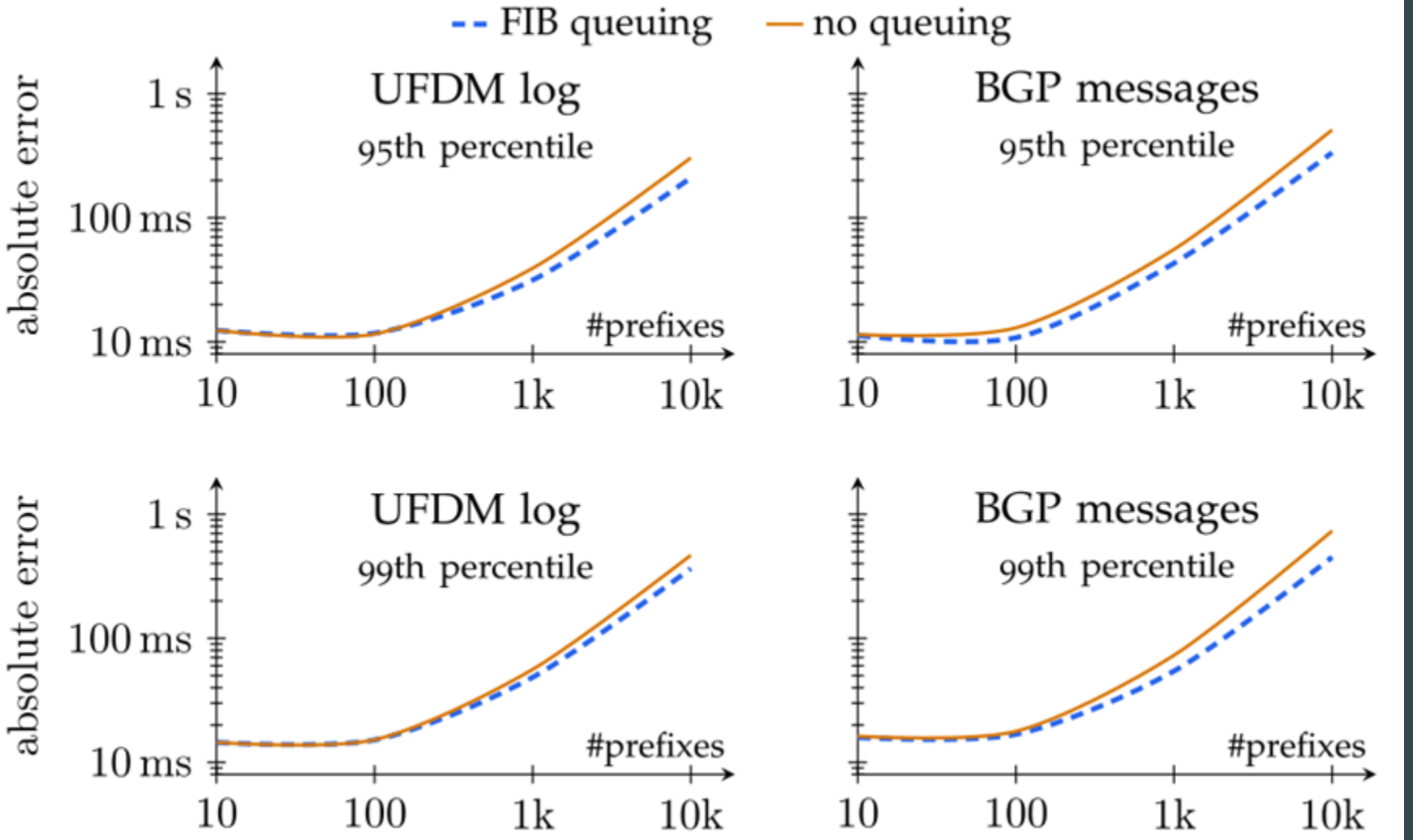


TRIX detects all long violations.

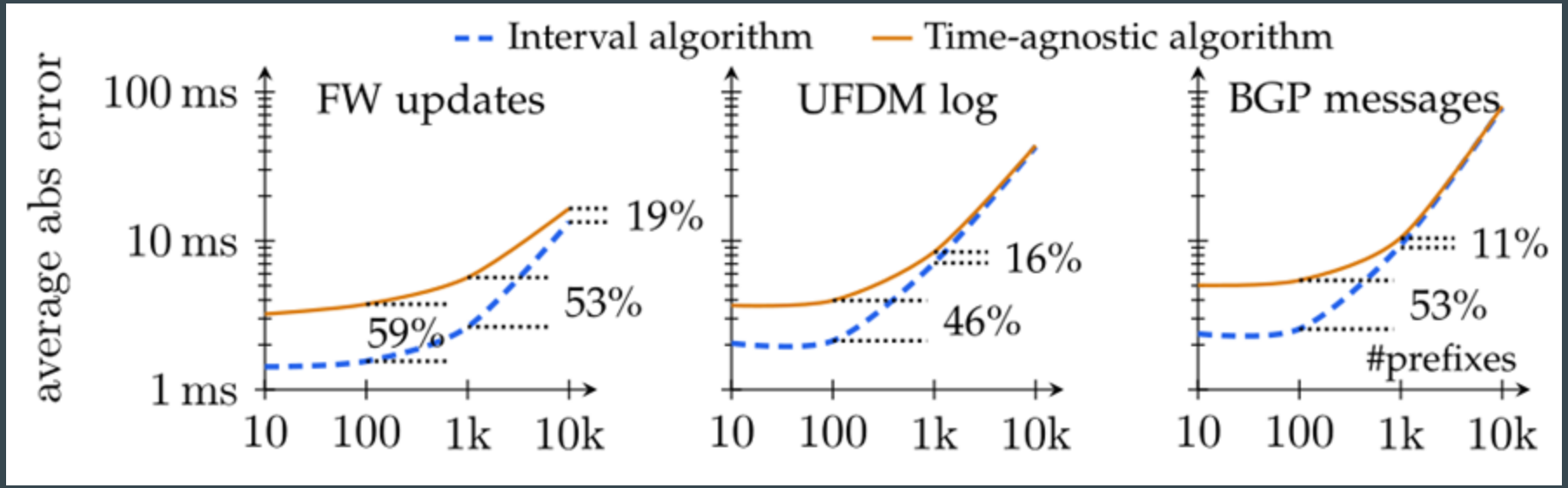


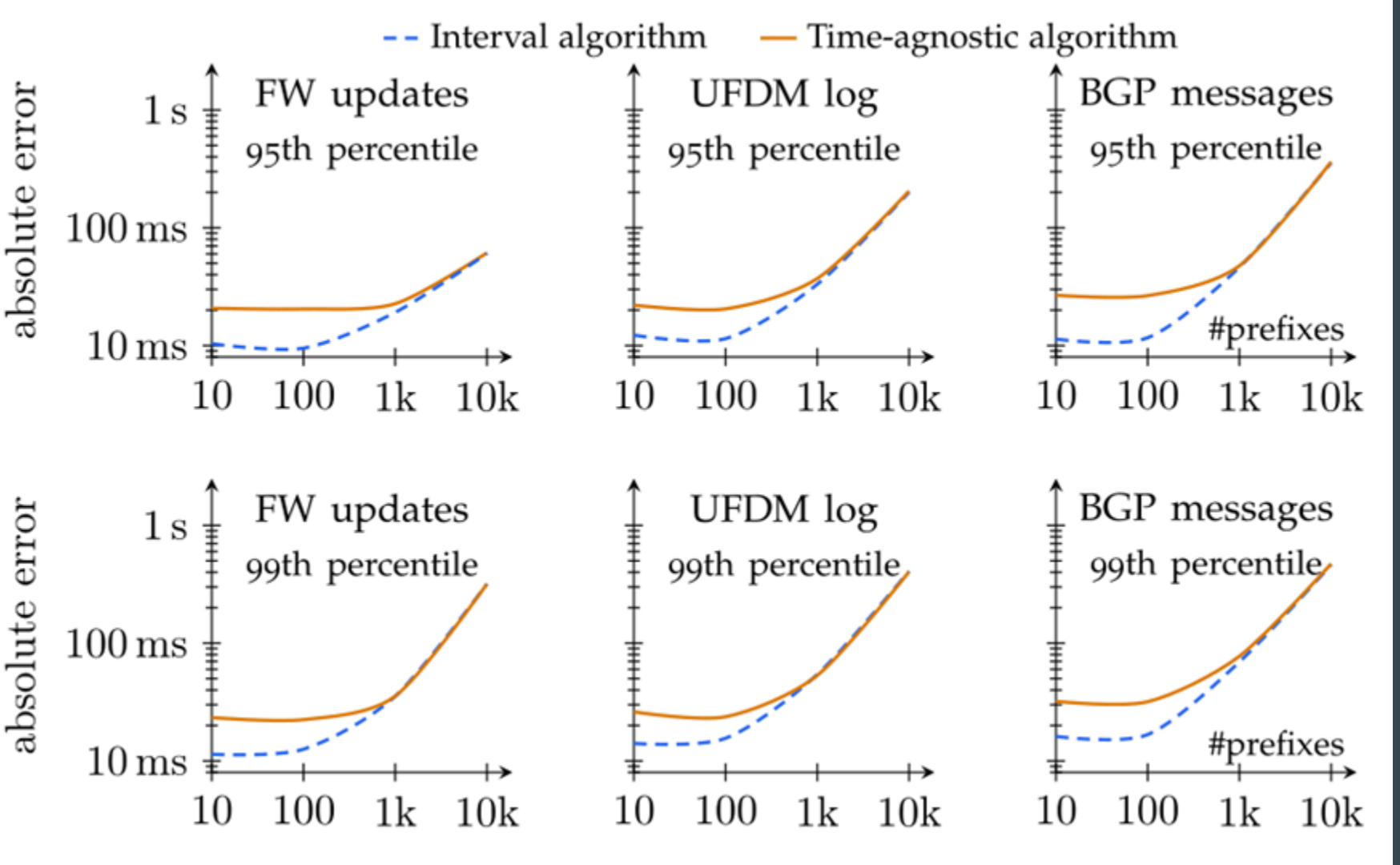
The FIB queuing model boosts TRIX' accuracy by 25.1% on average.




FIB queuing
no queuing
absolute error
1 s
100 ms
10 ms
UFDM log
95th percentile
#prefixes
10
100
1k
10k
BGP messages
95th percentile
#prefixes
10
100
1k
10k
absolute error
1 s
100 ms
10 ms
UFDM log
99th percentile
#prefixes
10
100
1k
10k
BGP messages
99th percentile
#prefixes
10
100
1k
10k

Modeling propagation improves TRIX' accuracy by up to 50%.




Interval algorithm
Time-agnostic algorithm
absolute error
FW updates
95th percentile
UFDM log
95th percentile
BGP messages
95th percentile
#prefixes
1 s
100 ms
10 ms
10
100
1k
10k
FW updates
99th percentile
UFDM log
99th percentile
BGP messages
99th percentile
#prefixes

TRIX computationally scales to large networks.