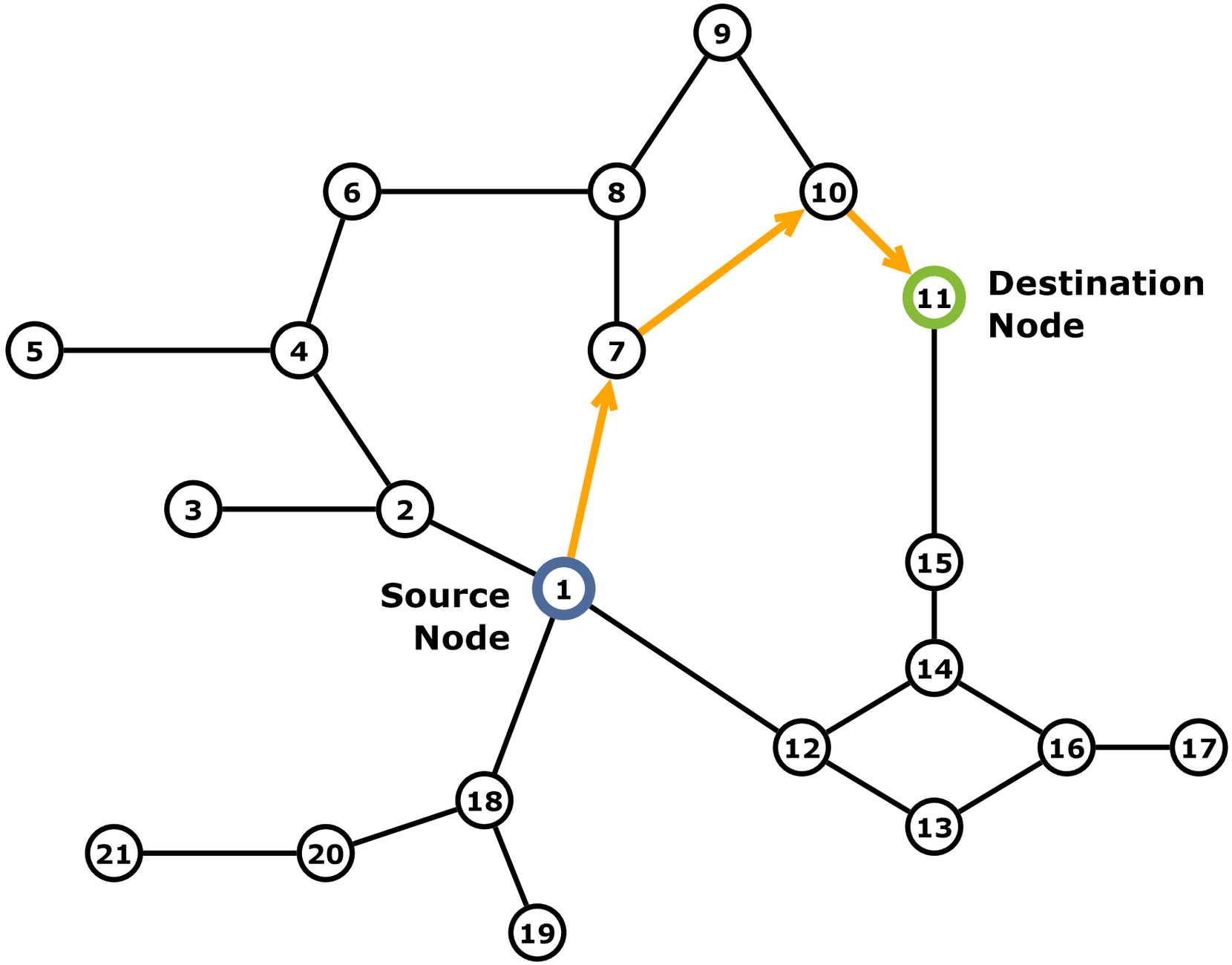
Grafting Arborescences for Extra Resilience of Fast Rerouting Schemes

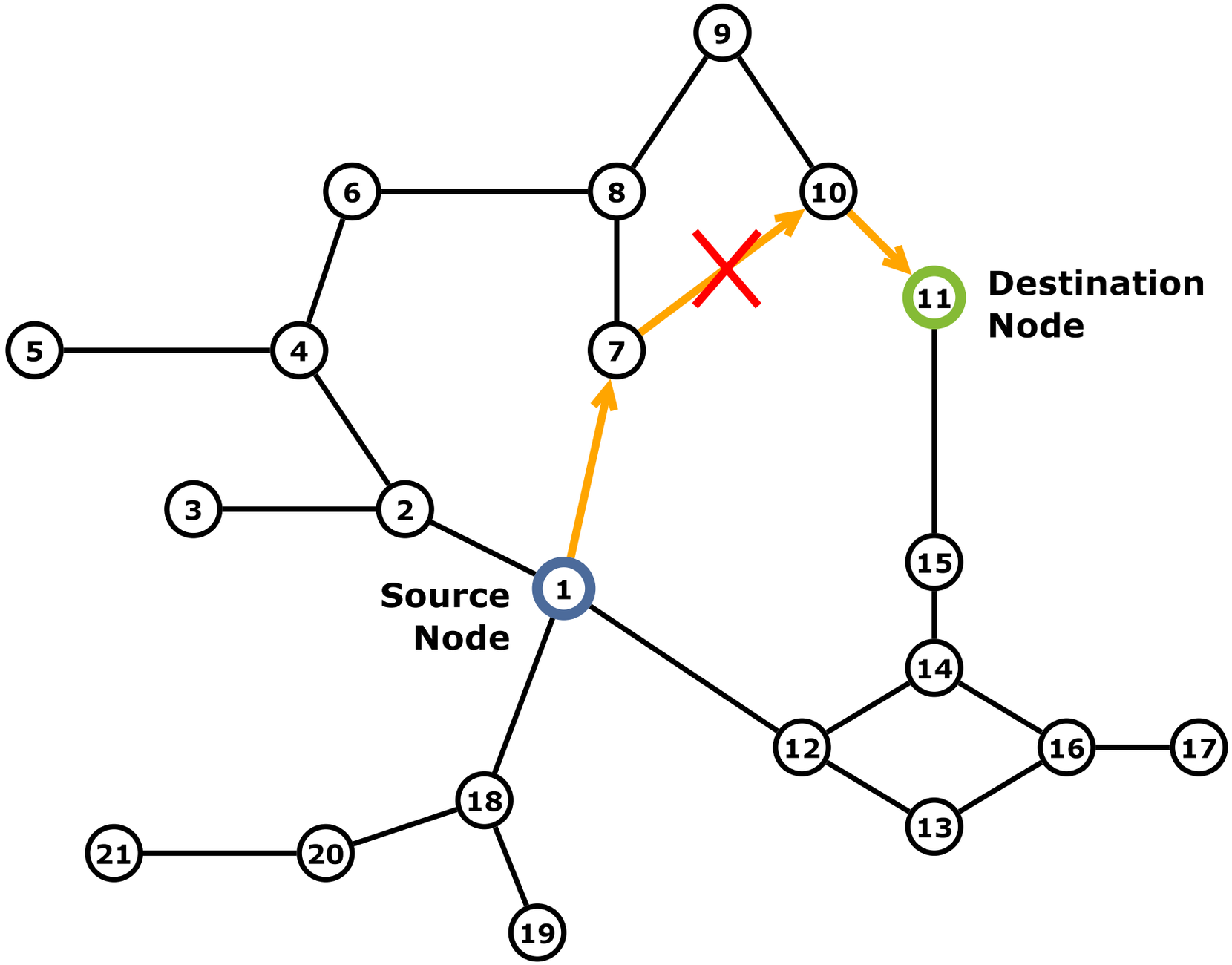
Klaus-Tycho Foerster (University of Vienna, Austria)
Andrzej Kamisiński (AGH University of Science and Technology in Kraków, Poland)
Yvonne-Anne Pignolet (DFINITY, Switzerland)
Stefan Schmid (University of Vienna, Austria)
Gilles Tredan (LAAS-CNRS, France)

IEEE INFOCOM 2021, Virtual Conference, 10-13 May 2021

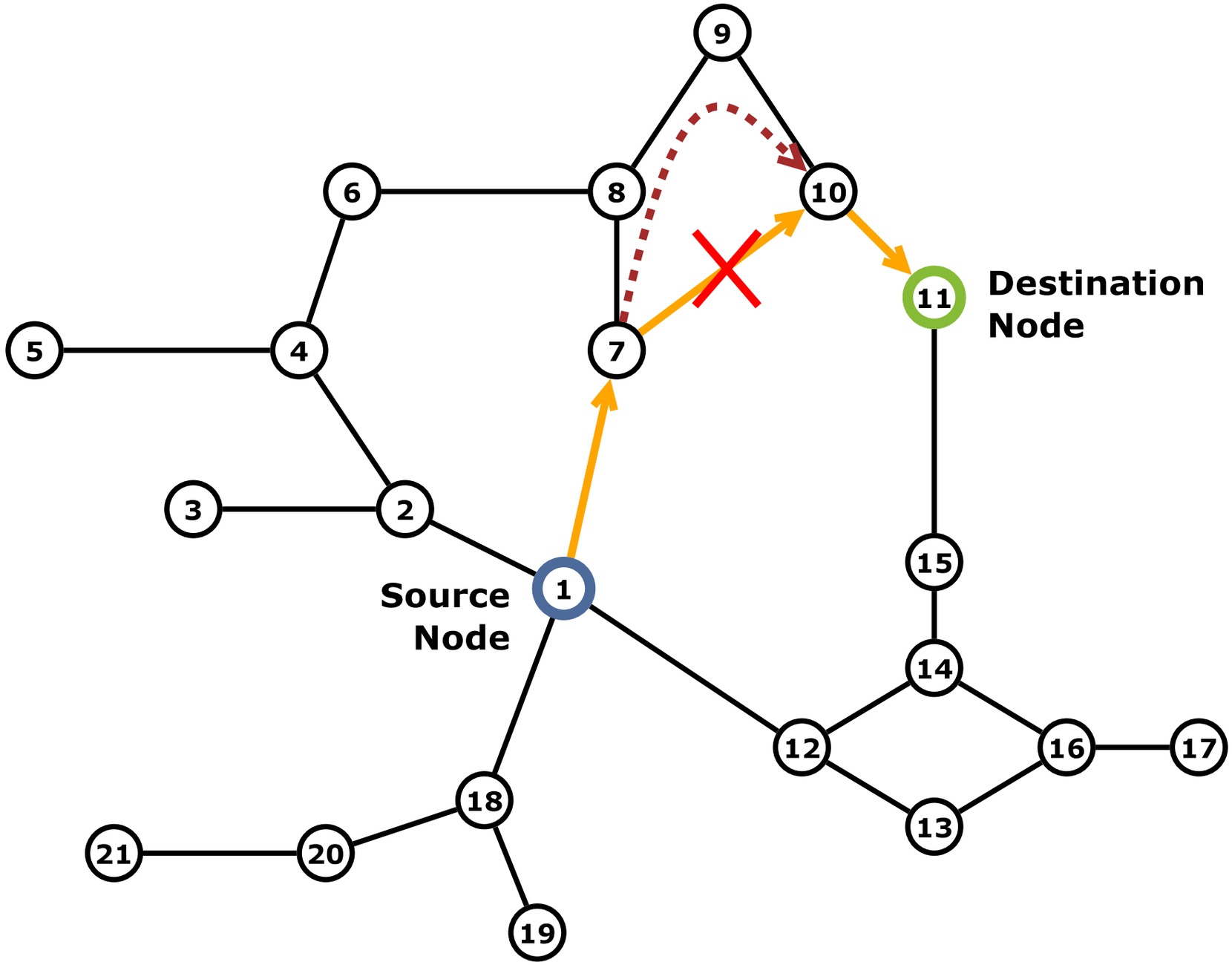
Introduction



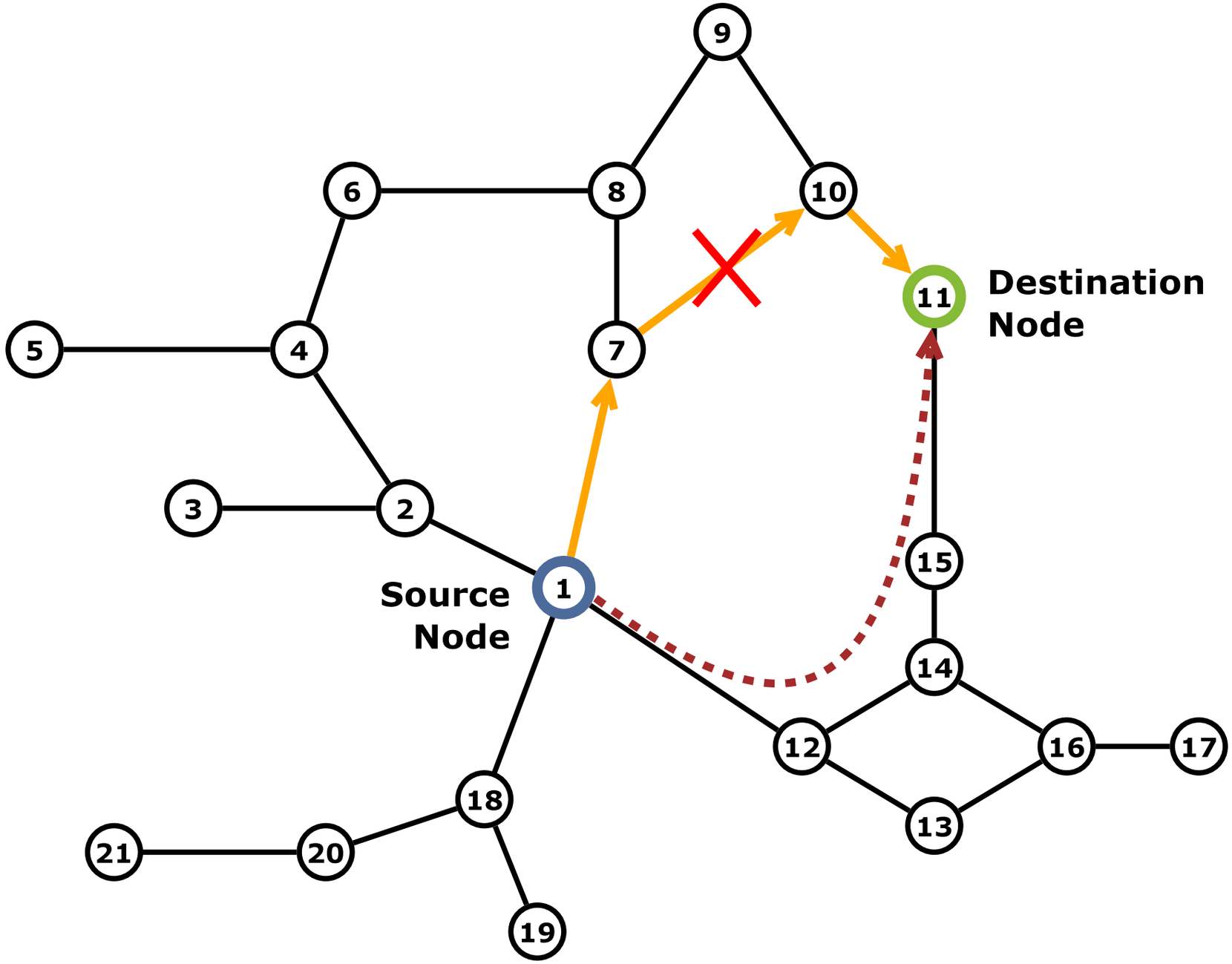
Introduction



Introduction



Introduction



How to deal with link/node failures?

Existing reconvergence mechanisms relying on the control-plane

- Signaling
- Relatively long recovery time
- Packet losses and increased delay

How to deal with link/node failures?

Existing reconvergence mechanisms relying on the control-plane

- Signaling
- Relatively long recovery time
- Packet losses and increased delay

Fast-Recovery mechanisms

- Based on locally-available information
- Low delay and packet losses
- **Examples:** Loop-Free Alternates, input interface-aware routing, tunneling
- **State of the art:**
  - Multiple arc-disjoint spanning arborescences

M. Chiesa, A. Kamisiński, J. Rak, G. Rétvári and S. Schmid, "A Survey of Fast-Recovery Mechanisms in Packet-Switched Networks," in *IEEE Communications Surveys & Tutorials*, 2021, doi: 10.1109/COMST.2021.3063980.

How to deal with link/node failures?

Existing reconvergence mechanisms relying on the control-plane

- Signaling
- Relatively long recovery time
- Packet losses and increased delay

Fast-Recovery mechanisms

- Based on locally-available information
- Low delay and packet losses
- **Examples:** Loop-Free Alternates, input interface-aware routing, tunneling
- **State of the art:**
  - Multiple arc-disjoint spanning arborescences
  - **Multiple extended/combined arborescences [THIS TALK]**

M. Chiesa, A. Kamisiński, J. Rak, G. Rétvári and S. Schmid, "A Survey of Fast-Recovery Mechanisms in Packet-Switched Networks," in *IEEE Communications Surveys & Tutorials*, 2021, doi: 10.1109/COMST.2021.3063980.

Motivation

Homogeneous networks: state-of-the-art algorithms based on redundant arc-disjoint arborescences

- Provide actual guarantees on recovery
- Do not scale well (example: 1-connected networks)

Heterogeneous networks: *Keep Forwarding*

- Shown to retain 99% reachability for 1/2/3 link failures
- No formal guarantee is claimed, not even for single link failures

Motivation

Homogeneous networks: state-of-the-art algorithms based on redundant arc-disjoint arborescences

- Provide actual guarantees on recovery
- Do not scale well (example: 1-connected networks)

Heterogeneous networks: *Keep Forwarding*

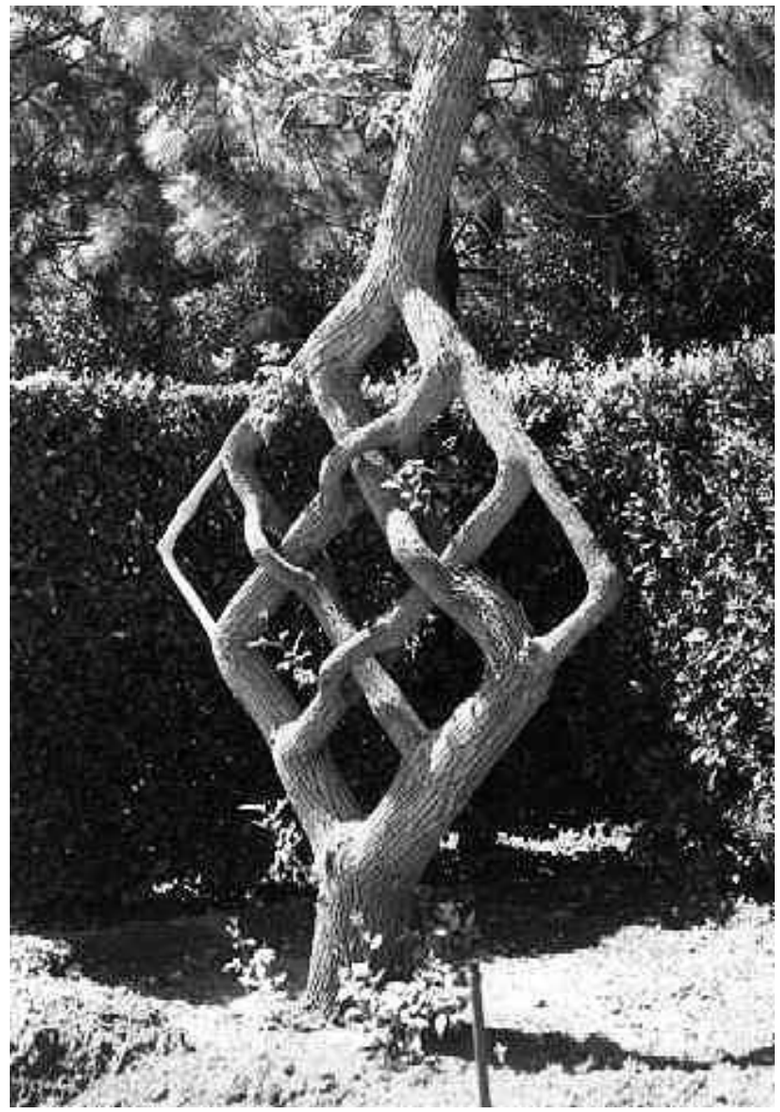
- Shown to retain 99% reachability for 1/2/3 link failures
- No formal guarantee is claimed, not even for single link failures

  We show that *Keep Forwarding* can fail after a single failure, even if the remaining network remains highly connected

Grafting: How to Build Better Arborescences for Heterogeneous Networks?

- **We extend or combine multiple arc-disjoint arborescences**, to be able to use more arcs (especially in densely connected networks)

- **We present three new fast-failover routing schemes that:**
  - Retain worst-case guarantees
  - Leverage network connectivity in non-adversarial multi-link failure scenarios



https://en.wikipedia.org

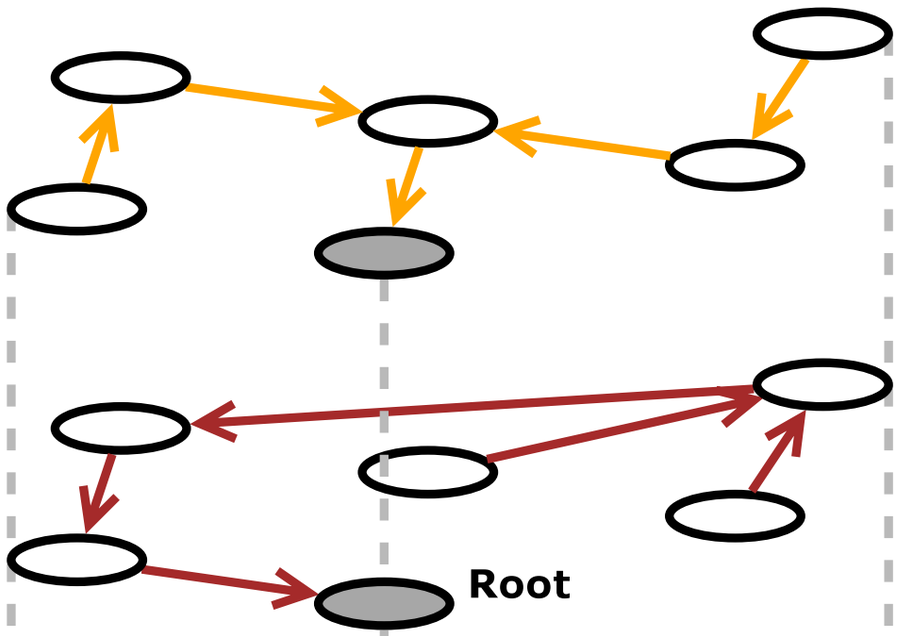
Outline

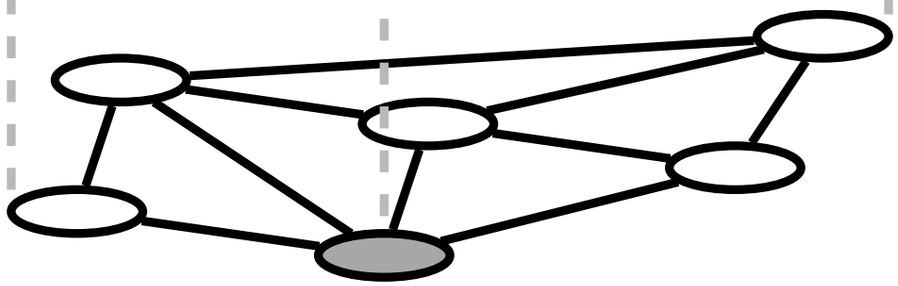
- The proposed algorithms
- The selected evaluation results
- Conclusion

DAG-FRR: Leveraging Network Heterogeneity via DAGs

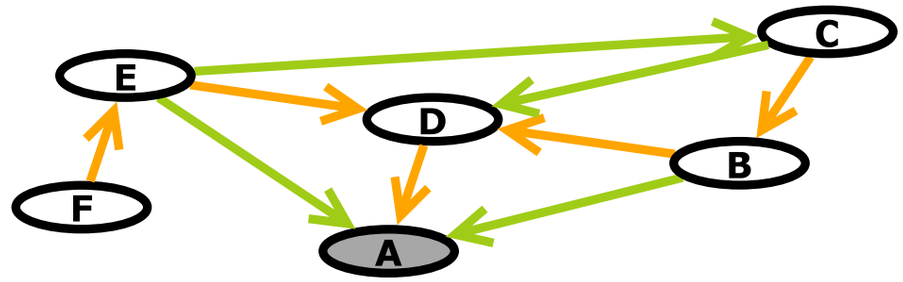
PART I

Decomposition (spanning arborescences)





PART II



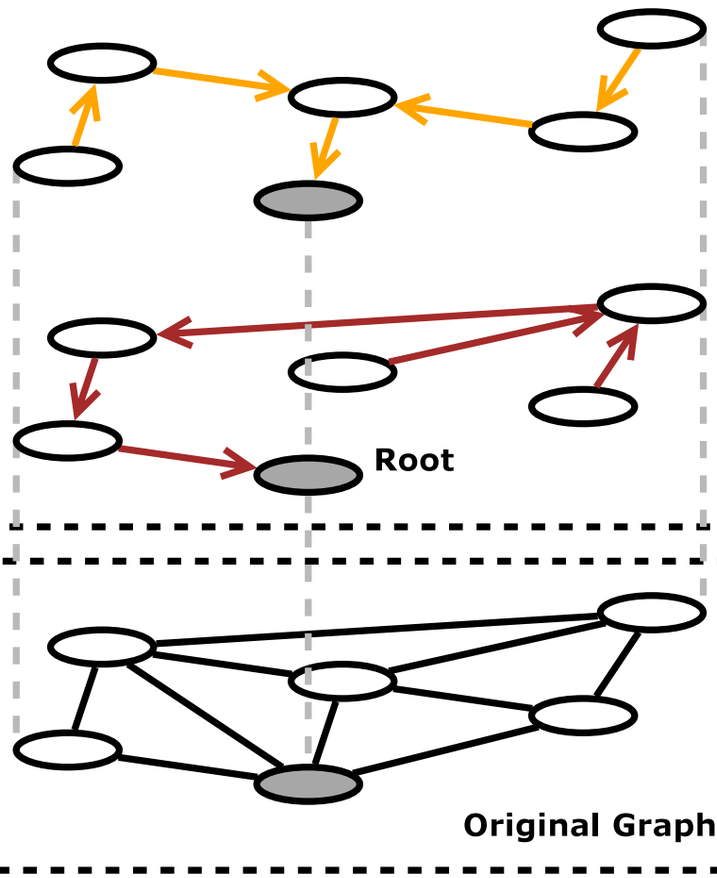
Extending arborescences to DAGs:
1) Unused arcs, one endpoint in DAG
2) Avoid cycles

PART III: Extending Arborescence Routing

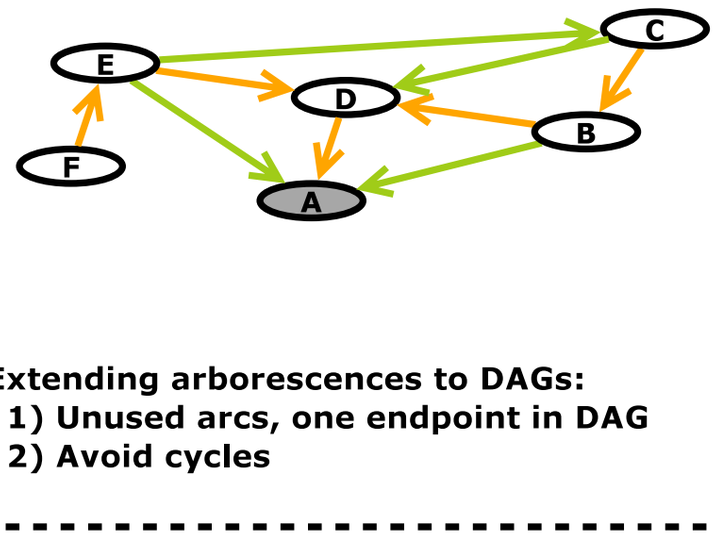
Example:
1) Choose an active link (of a DAG) reducing the distance to the root the most
2) If all candidate links have failed, switch to the next DAG

DAG-FRR: Leveraging Network Heterogeneity via DAGs

PART I



PART II



Extending arborescences to DAGs:
1) Unused arcs, one endpoint in DAG
2) Avoid cycles

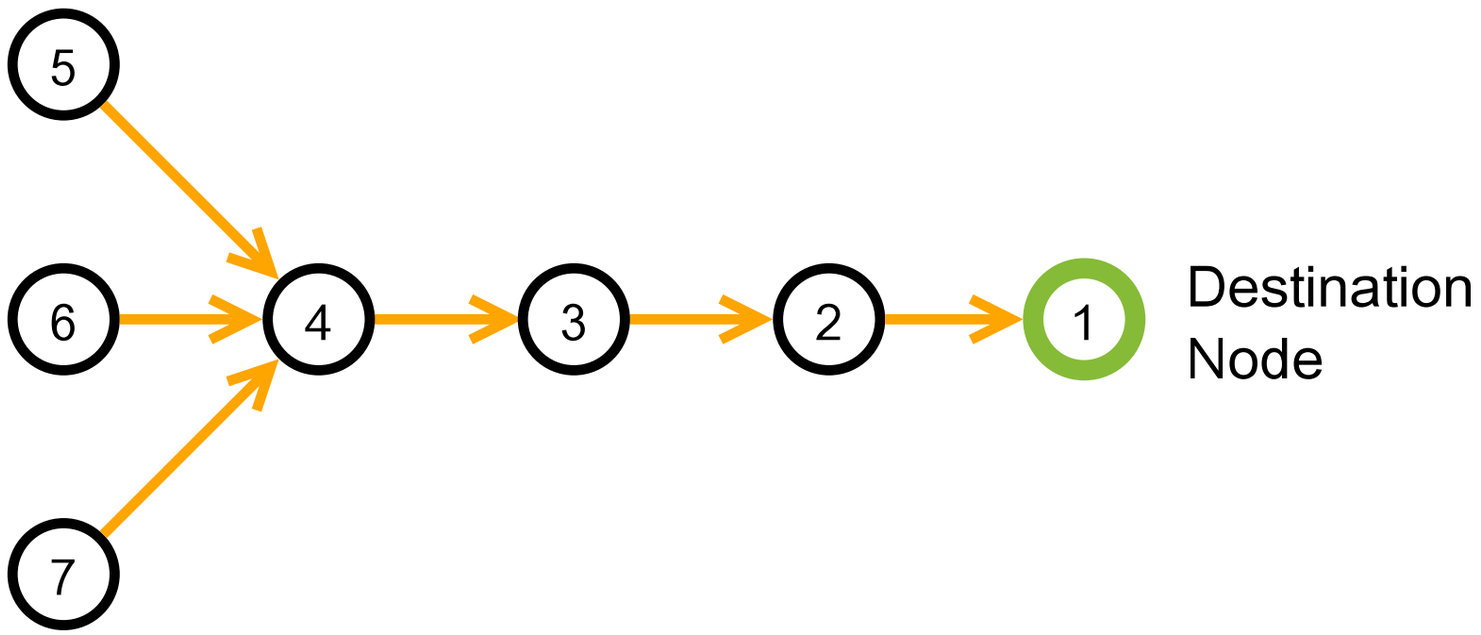
PART III: Extending Arborescence Routing

Example:
1) Choose an active link (of a DAG) reducing the distance to the root the most
2) If all candidate links have failed, switch to the next DAG

Starting on *k* arc-disjoint rooted spanning arborescences/DAGs, *DAG-FRR* is resilient to *k*—1 arc failures

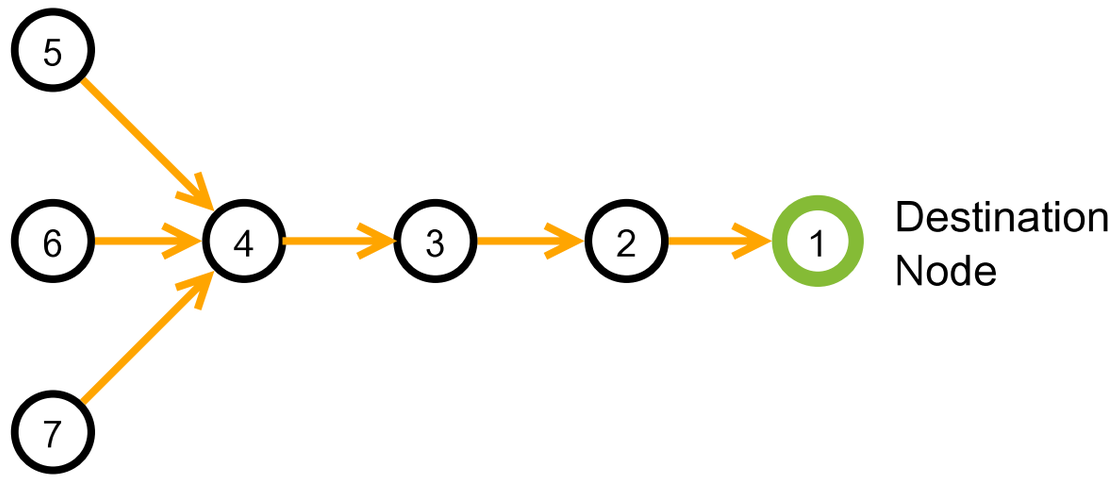
Cluster-FRR: Cluster-Based Forest Construction

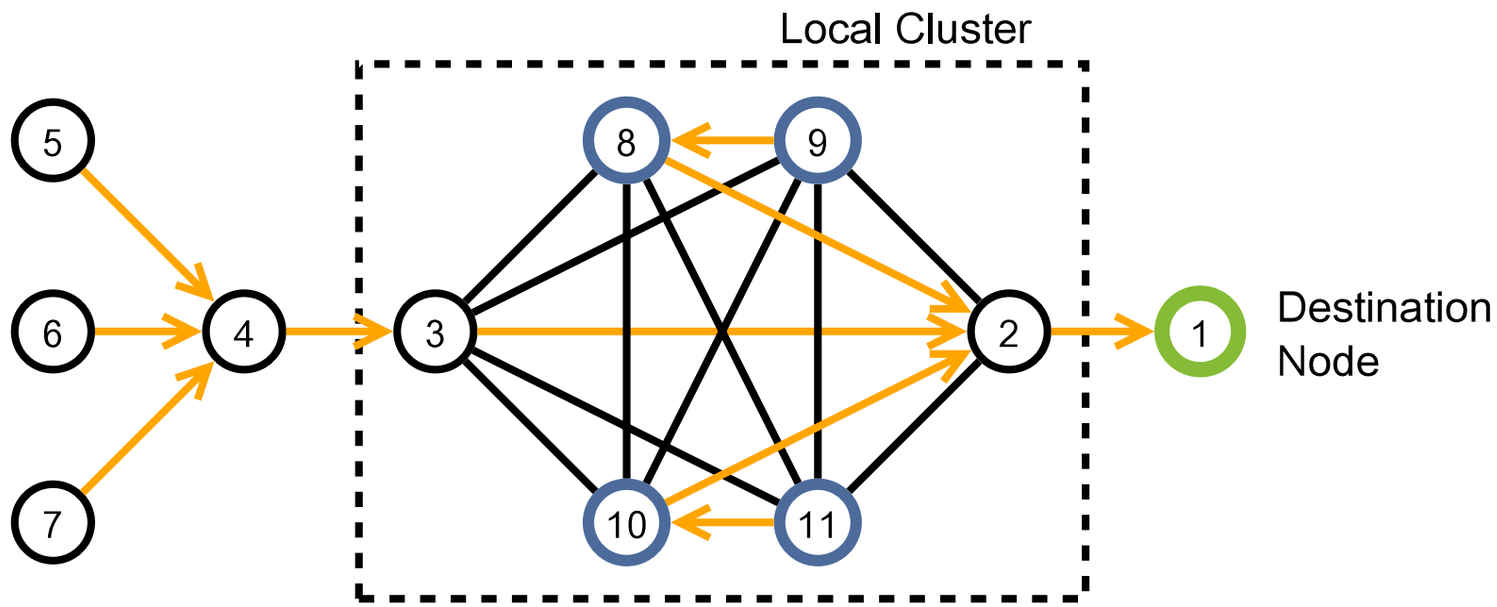
PART I: Decomposition
(1-connected graph)



Cluster-FRR: Cluster-Based Forest Construction

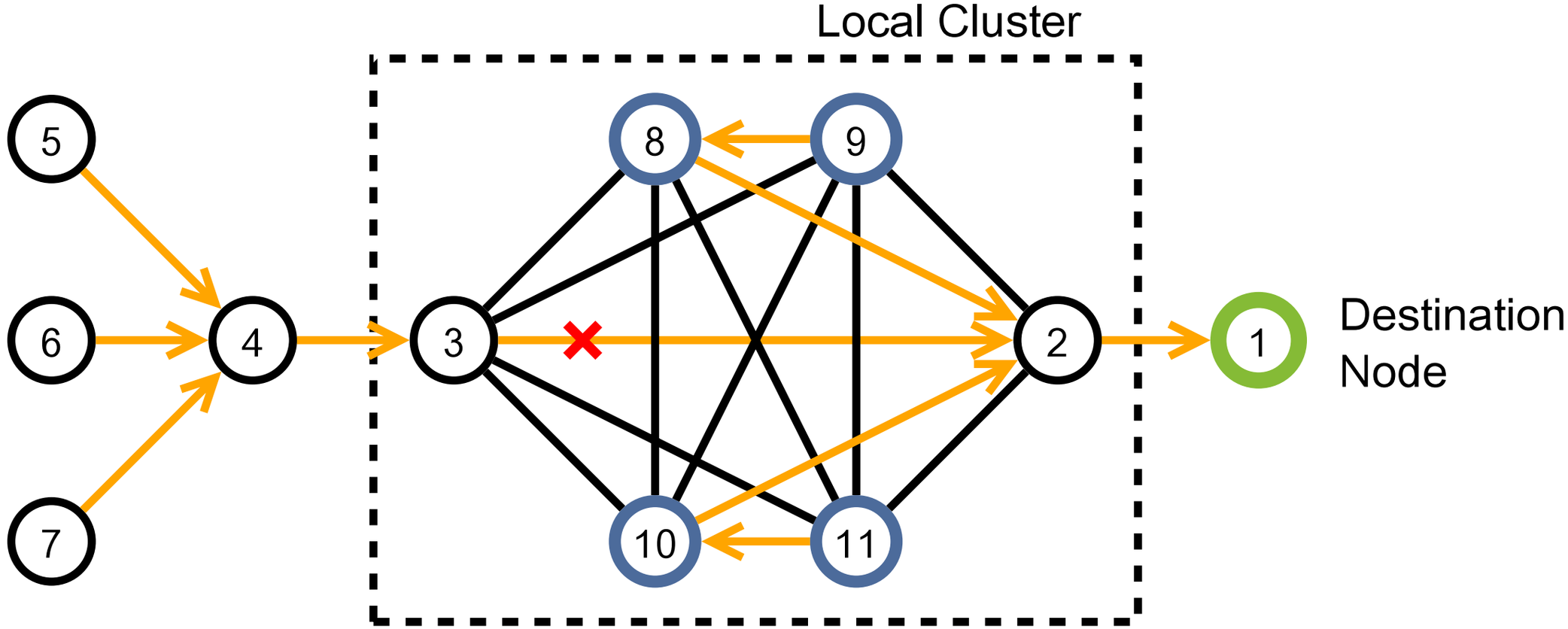
PART I: Decomposition
(1-connected graph)





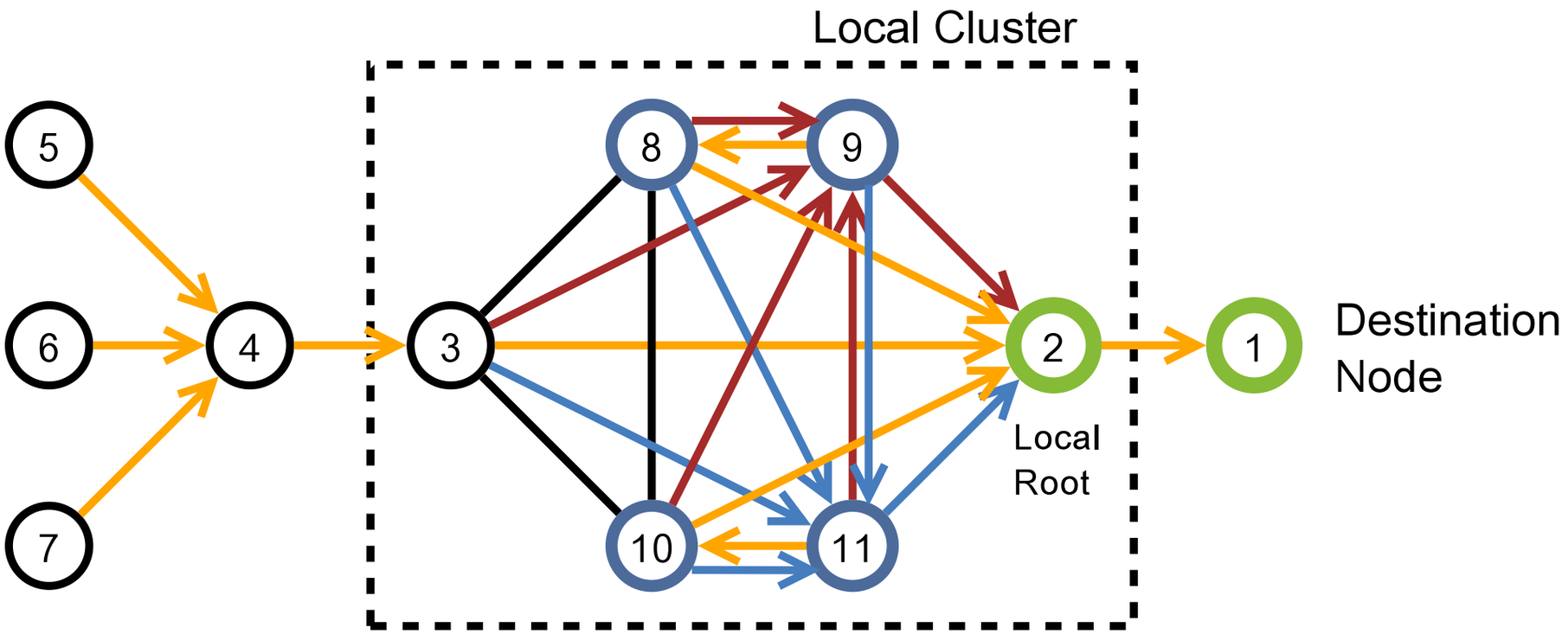
Cluster-FRR: Cluster-Based Forest Construction

PART I: Decomposition
(1-connected graph)



Cluster-FRR: Cluster-Based Forest Construction

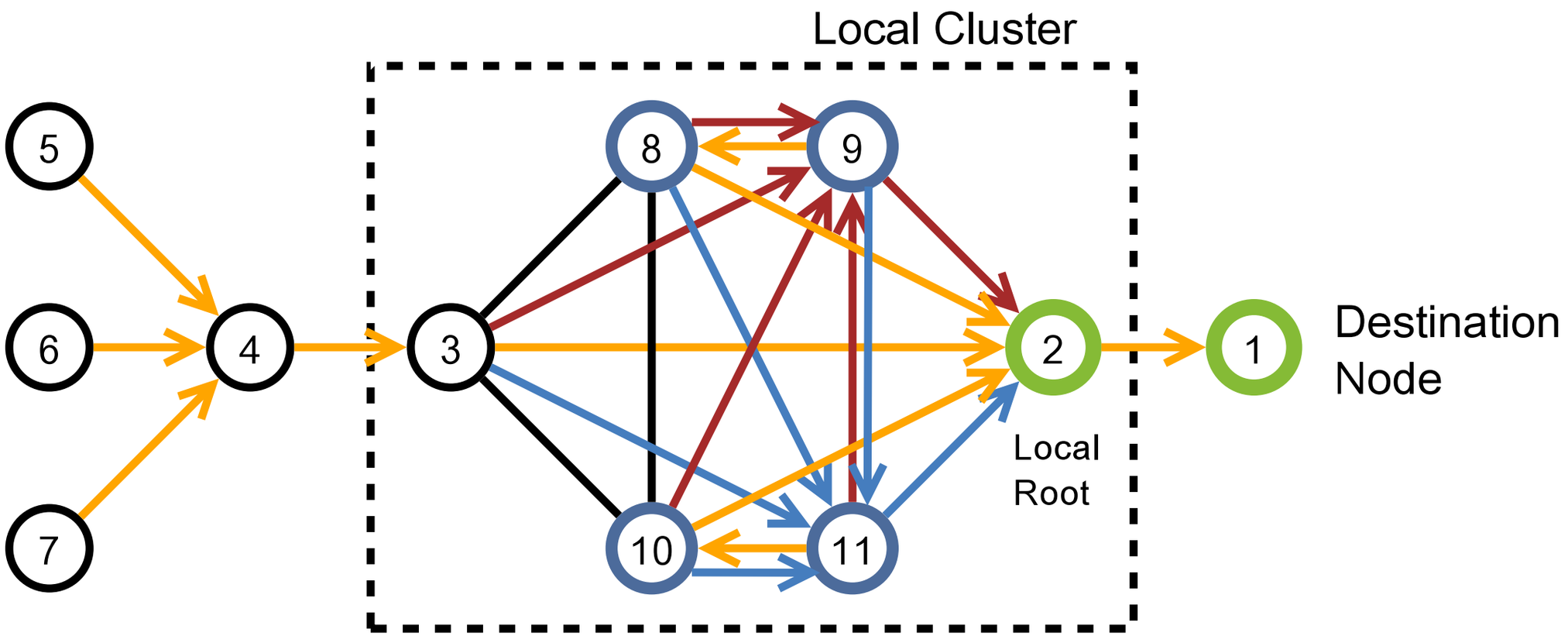
PART II: Cluster-Based Forest Construction



1) Find local clusters based on the value of the clustering coefficient
2) Within each cluster, select the root node as the closest node to the destination
3) Given the local root, find arc-disjoint spanning arborescences within the cluster
4) Exclude arcs belonging to primary arborescences from the local ones

Cluster-FRR: Cluster-Based Forest Construction

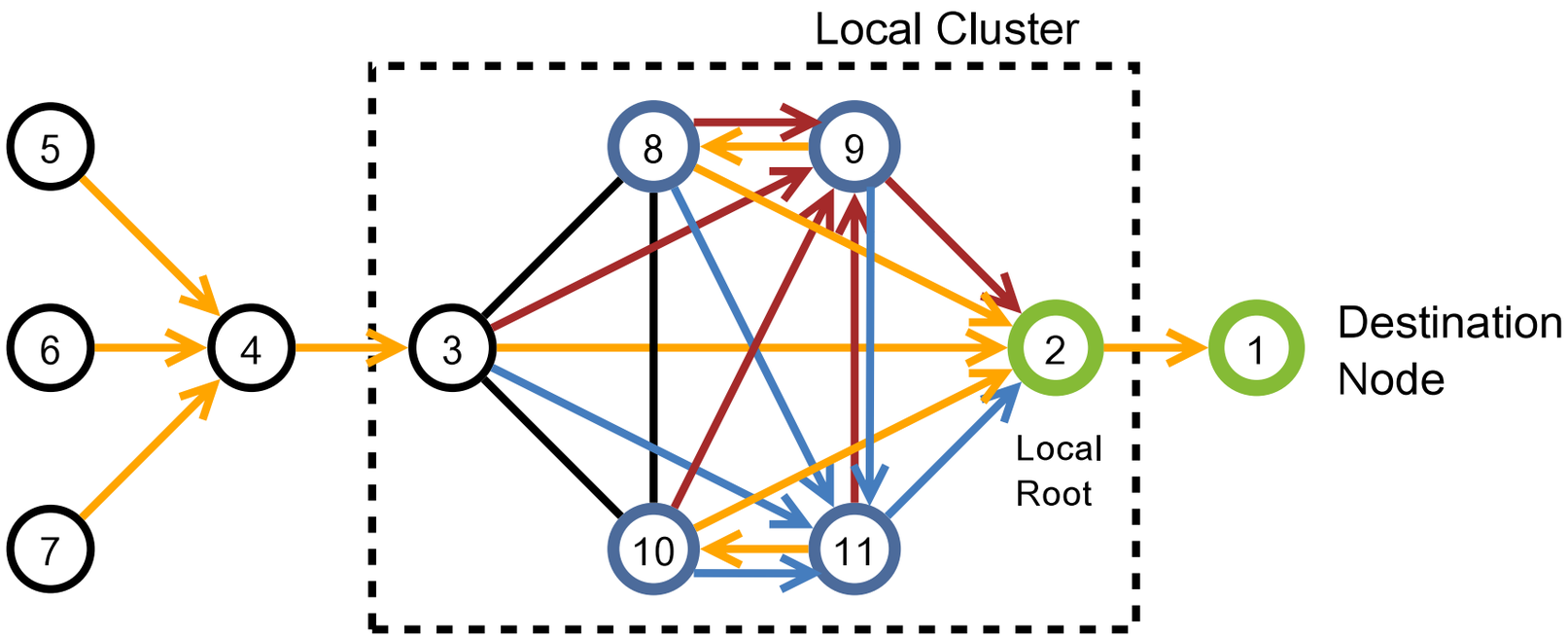
PART III: Extending Arborescence Routing



1) If possible, use existing spanning arborescences (circular routing)
2) Otherwise, switch to partial arborescences in local clusters

Cluster-FRR: Cluster-Based Forest Construction

PART III: Extending Arborescence Routing

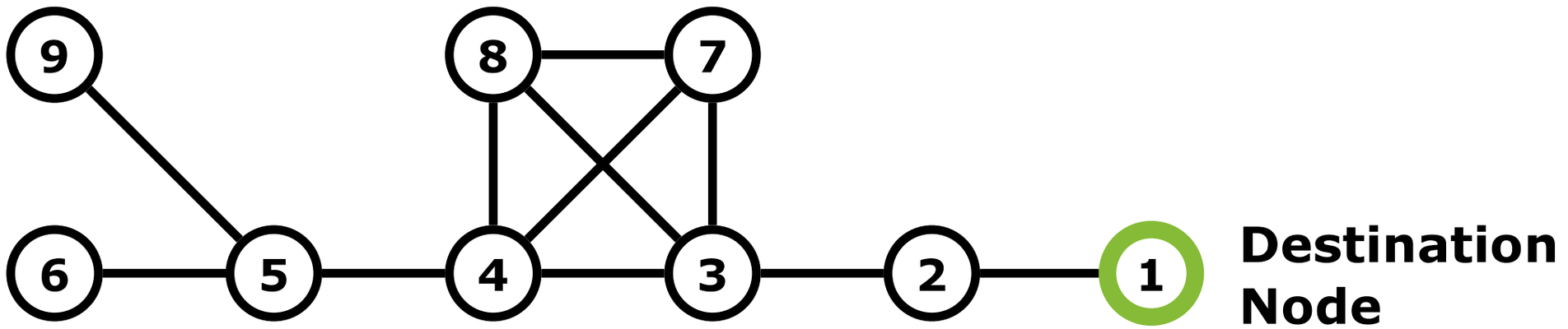


1) If possible, use existing spanning arborescences (circular routing)
2) Otherwise, switch to partial arborescences in local clusters

On *k*-connected graphs, *Cluster-FRR* is resilient to *k*—1 arc failures

Augment-FRR: Connectivity Augmentation

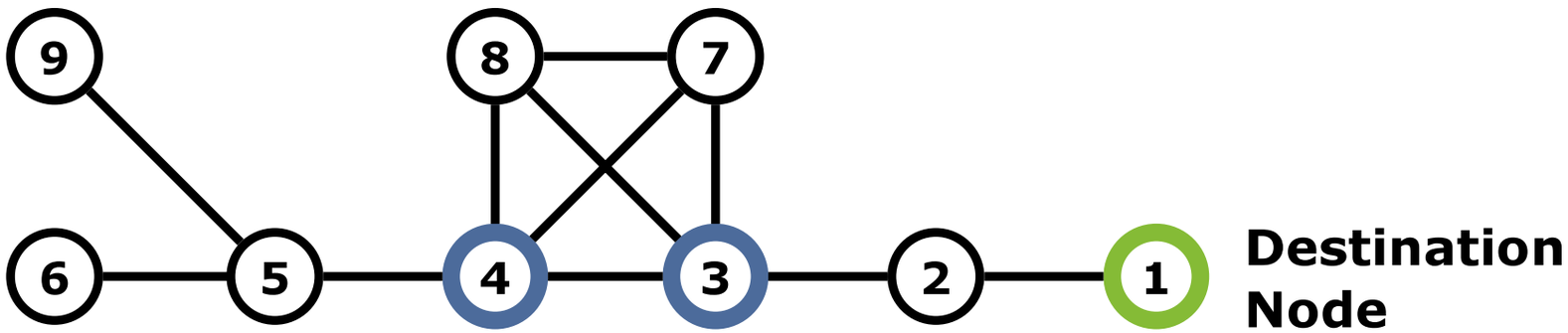
PART I: Graph Augmentation (virtual links)

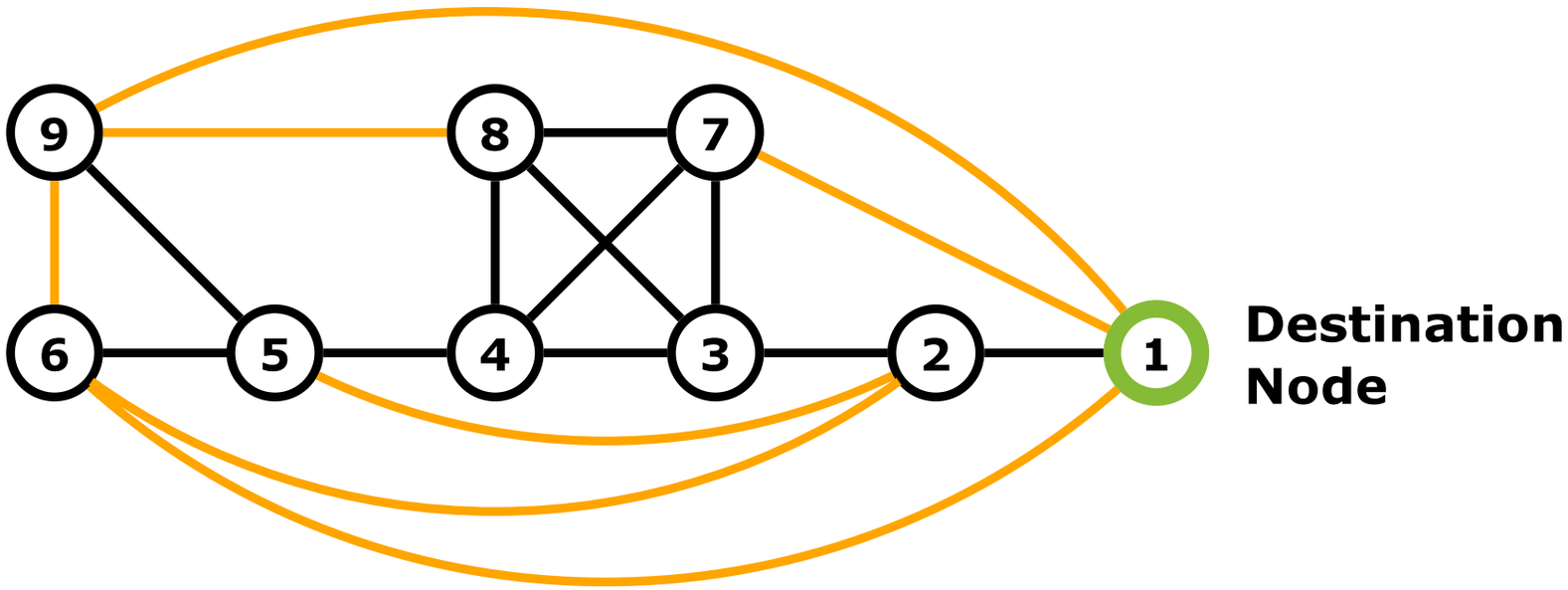


1) Select the node with the highest degree d
2) Augment the network to be d-connected, using as few virtual links as possible

Augment-FRR: Connectivity Augmentation

PART I: Graph Augmentation (virtual links)





Augment-FRR: Connectivity Augmentation

PART II: Building Arborescences on Virtual Links

- **Question:** How to distribute real and virtual links over the arborescences?
- **Answer:** Run a greedy rooted spanning arborescence decomposition algorithm

While growing arborescence T_i:

- Add a candidate arc a **only if** the remaining network (excluding arc a and arborescences $T_1, T_2, \ldots, T_i$) is still $(d\text{-}i)$—connected
- Prefer arcs associated with **real** links

Augment-FRR: Connectivity Augmentation

PART II: Building Arborescences on Virtual Links

- **Question:** How to distribute real and virtual links over the arborescences?
- **Answer:** Run a greedy rooted spanning arborescence decomposition algorithm

While growing arborescence T_i:

- Add a candidate arc *a* **only if** the remaining network (excluding arc *a* and arborescences $T_1, T_2, \ldots, T_i$) is still $(d\text{-}i)$—connected
- Prefer arcs associated with **real** links

On *k*-connected graphs, *Augment-FRR* is resilient to *k*—1 arc failures

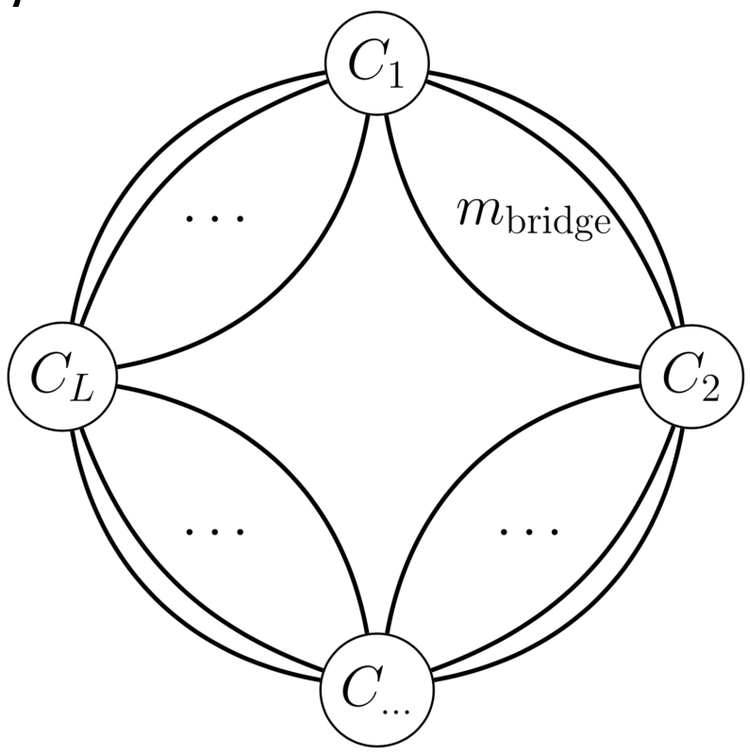
Evaluation

Link failure models

- Random link failures
- Targeted attacks (model based on local clusters)

Network topologies

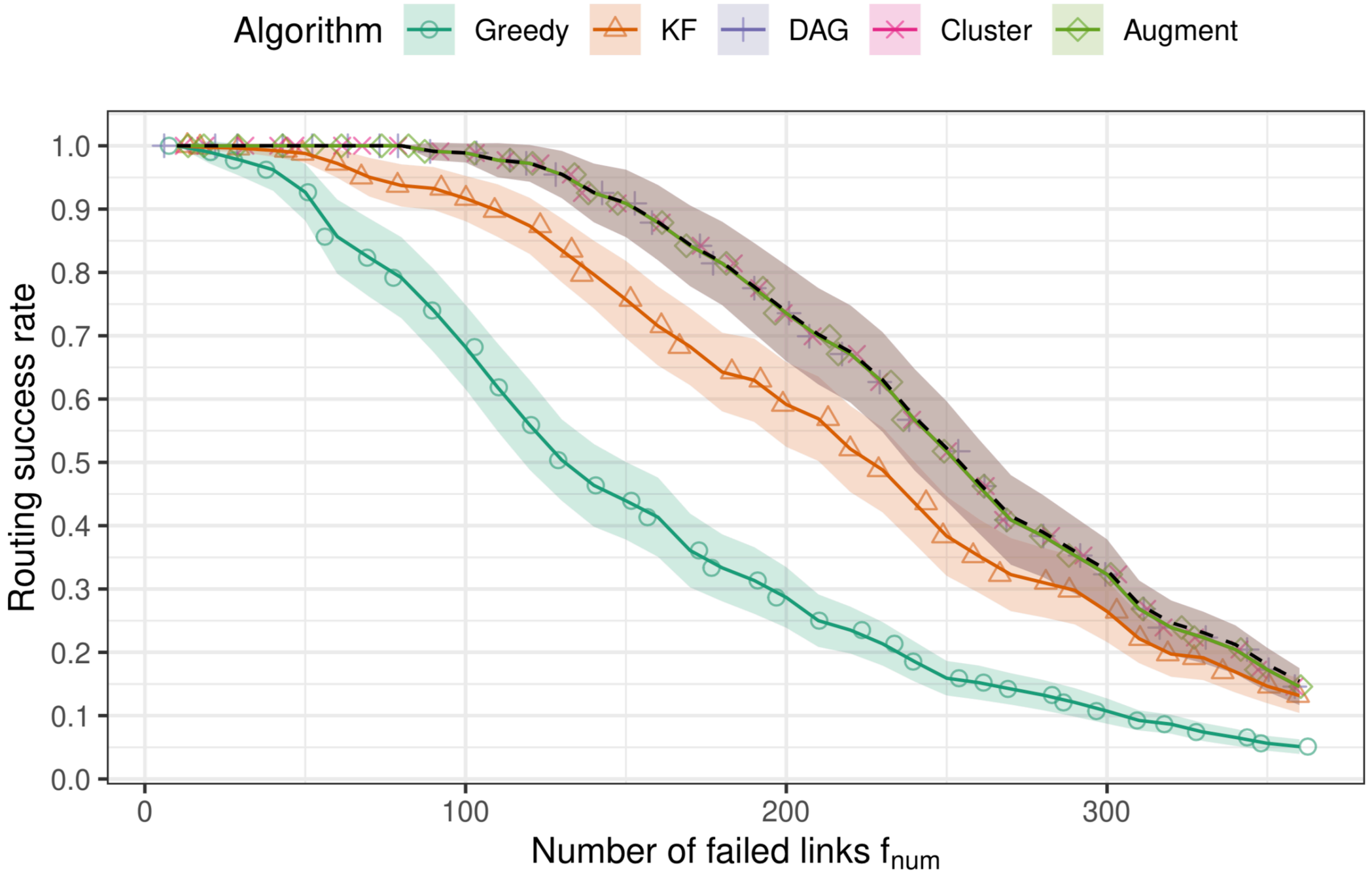
- Ring-of-cliques (synthetic; $k_{\text{clique}} > k_{\text{graph}}$)
- Topology Zoo (real-world networks)



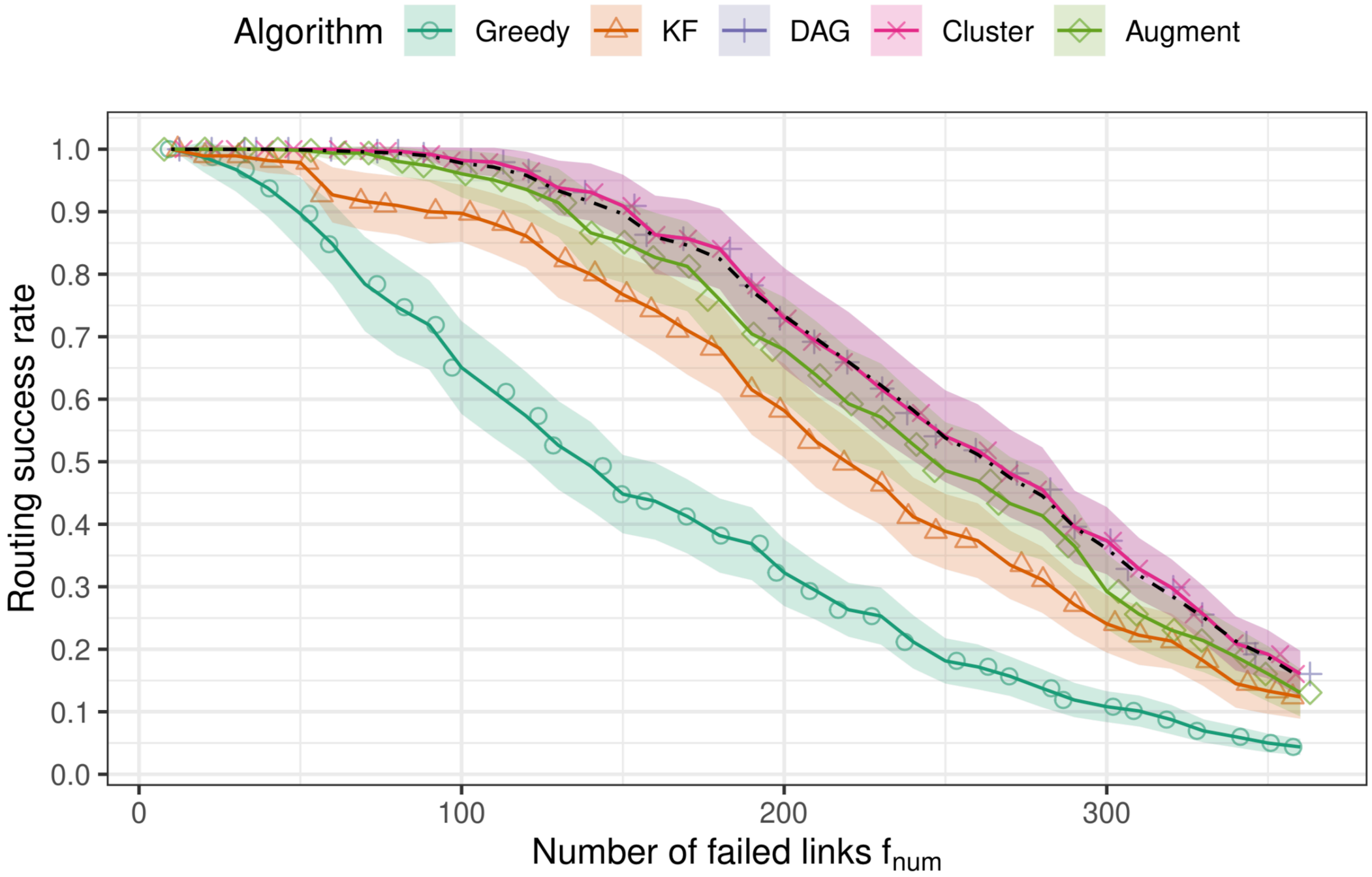
Extensive simulations

- Routing success rate
- Fraction of connected nodes after failures
- Time to compute the routing tables

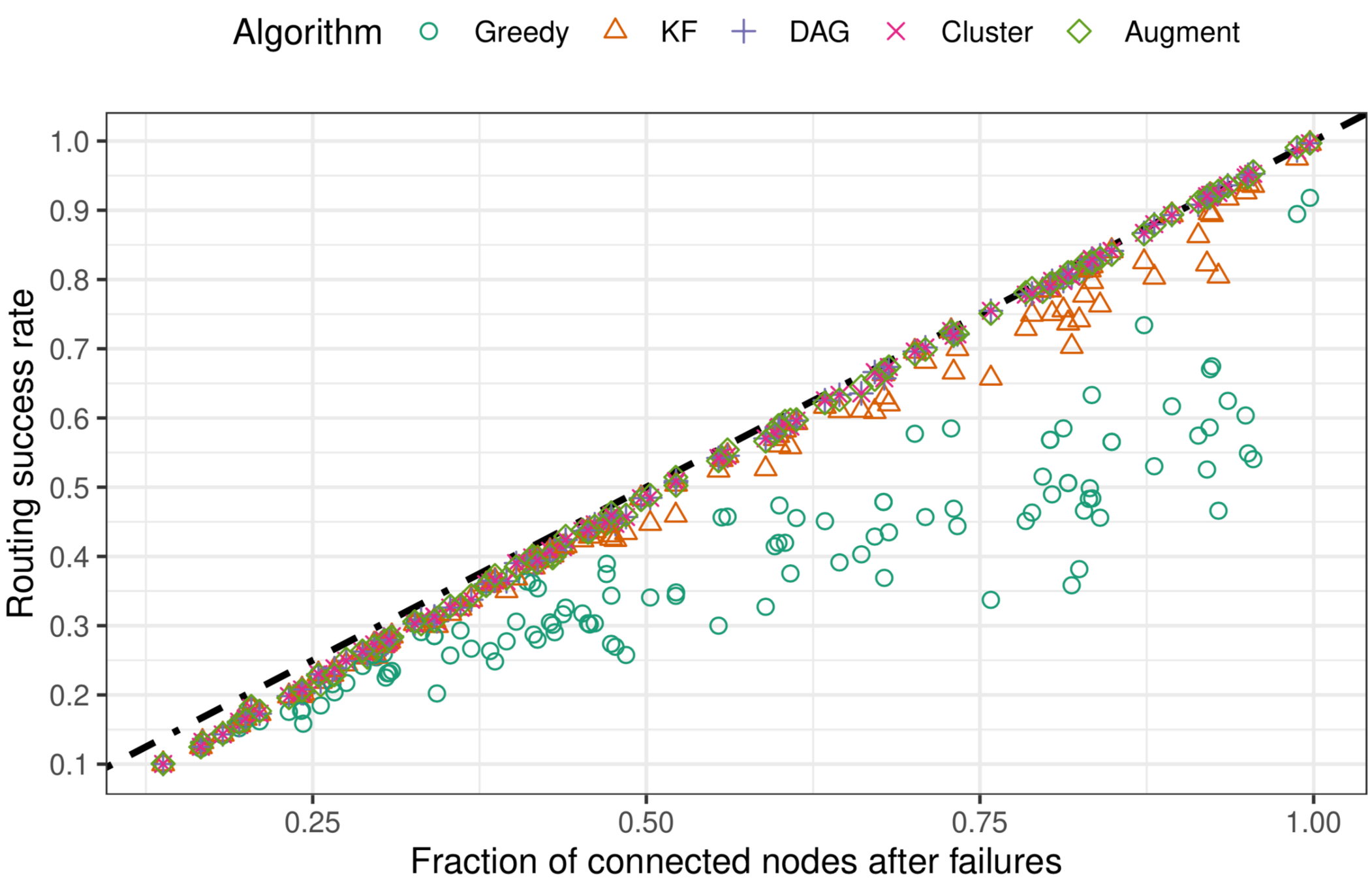
Routing Success Rate in Highly Heterogeneous Networks (RANDOM)



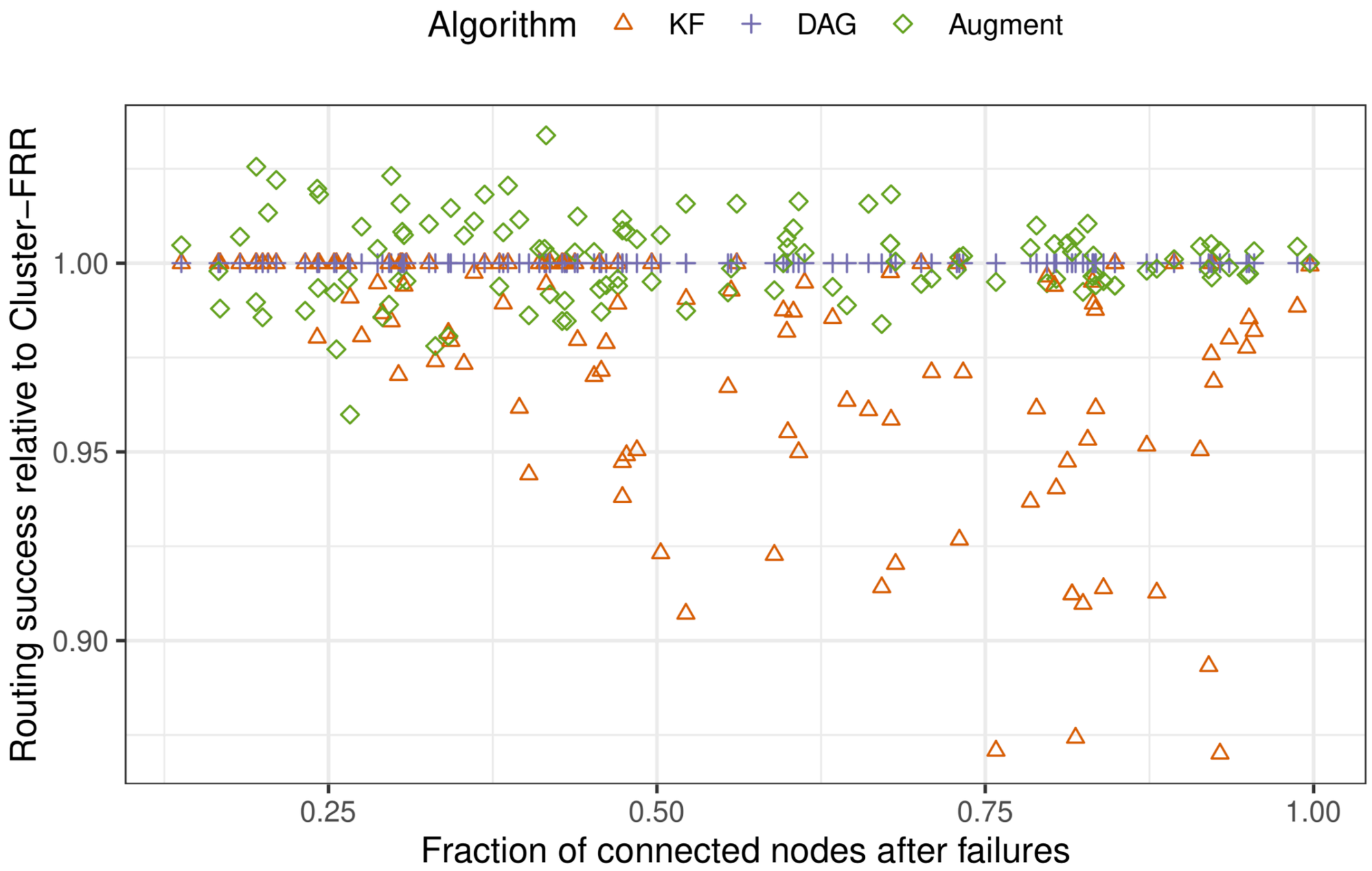
Routing Success Rate in Highly Heterogeneous Networks (CLUSTER)



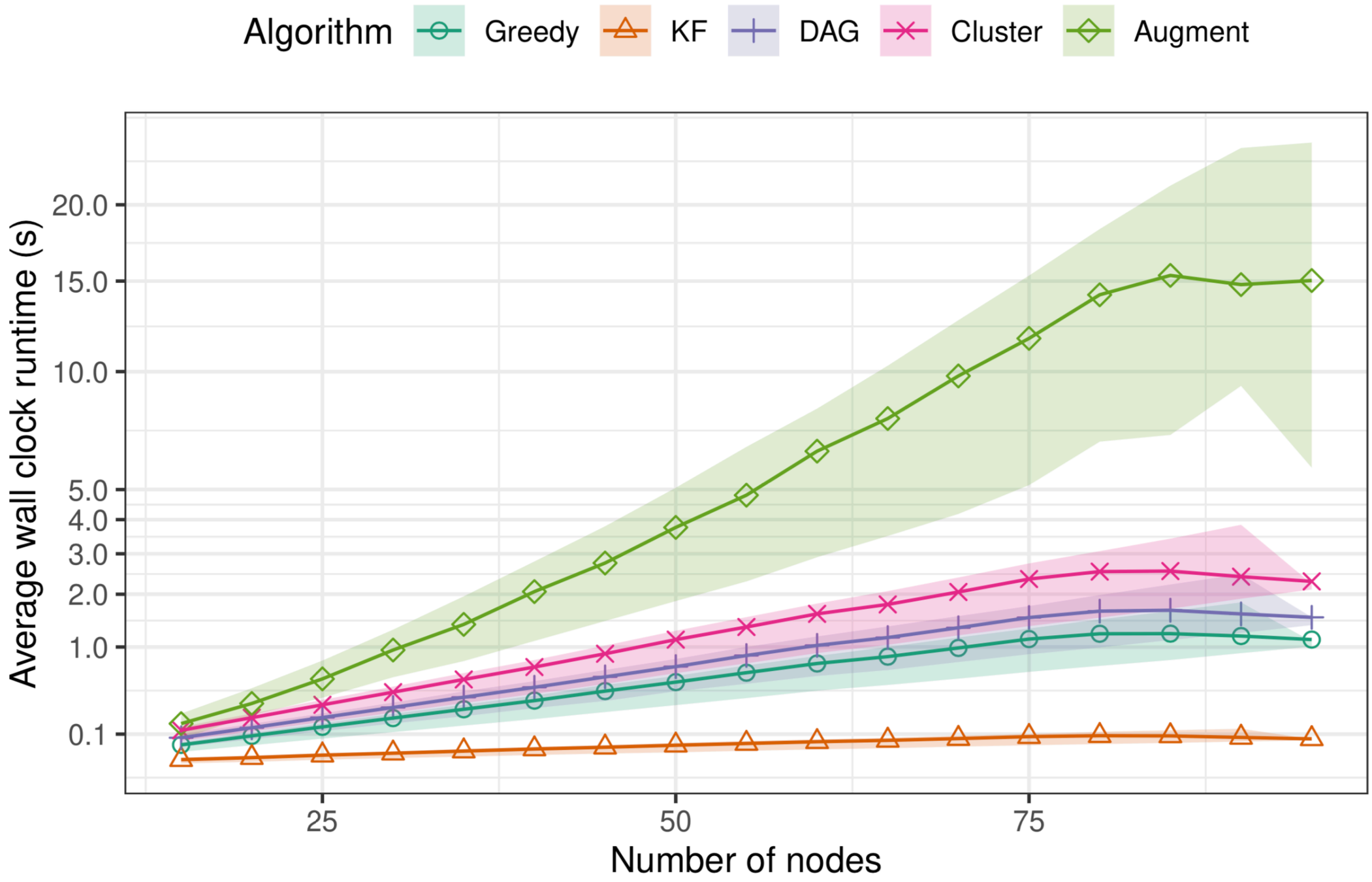
Routing Success Rate on Real-World Networks (RANDOM)



Relative Routing Success Rate on Real-World Networks (RANDOM)



Precomputation of the Routing Tables: Average Wall Clock Runtime



Conclusion

The proposed algorithms:

- **Maintain the worst-case guarantees** of state-of-the-art algorithms based on spanning-arborescences
- **Leverage the structure of heterogeneous networks** for non-adversarial failure scenarios
- **Outperform state-of-the-art solutions** in that they allow to maintain connectivity under significantly more concurrent link failures

The source code and simulation results will soon be available at:
https://gitlab.cs.univie.ac.at/ct-papers/fast-failover

Thank you for your attention

Grafting Arborescences for Extra Resilience of Fast Rerouting Schemes

Klaus-Tycho Foerster (University of Vienna, Austria)
Andrzej Kamisiński (AGH University of Science and Technology in Kraków, Poland)
Yvonne-Anne Pignolet (DFINITY, Switzerland)
Stefan Schmid (University of Vienna, Austria)
Gilles Tredan (LAAS-CNRS, France)

Presenter: Andrzej Kamisiński (andrzejk@agh.edu.pl)