How to Establish Loop-Free Multipath Routes in Named Data Networking?

NDNcomm 2017

Klaus Schneider, Beichuan Zhang

March 24, 2017

The University of Arizona

Routing and Forwarding in IP Networks

No Loop Detection on Forwarding Plane

- Very high cost (packets loop through same routers until TTL runs out)

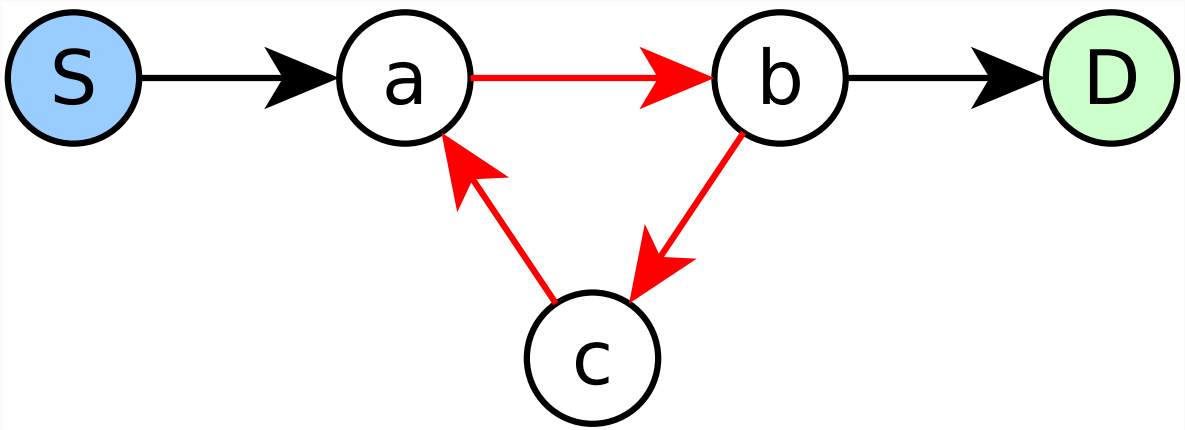
$\Rightarrow$ Routing has to be perfectly **loop-free**

Routing and Forwarding in IP Networks

No Loop Detection on Forwarding Plane

- Very high cost (packets loop through same routers until TTL runs out)

$\Rightarrow$ Routing has to be perfectly **loop-free**

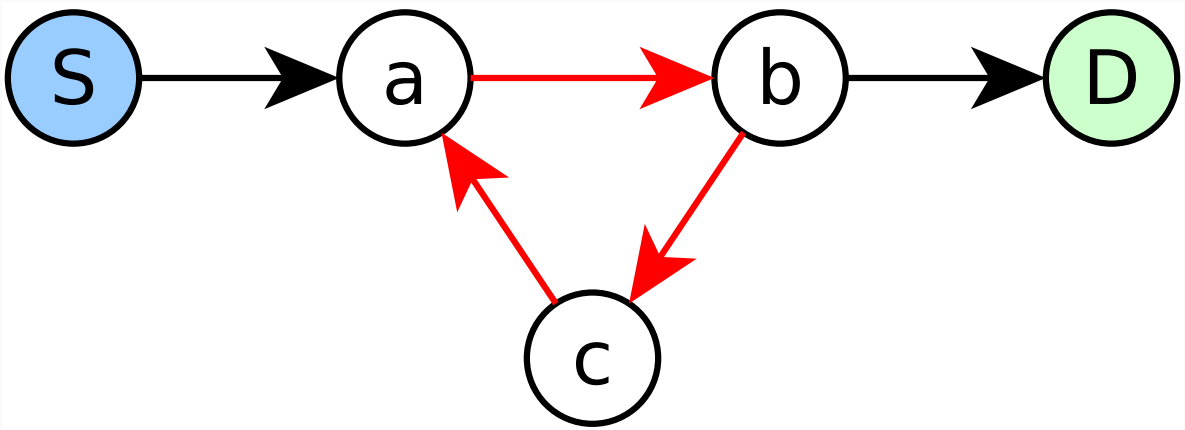


Routing and Forwarding in IP Networks

No Loop Detection on Forwarding Plane

- Very high cost (packets loop through same routers until TTL runs out)

$\Rightarrow$ Routing has to be perfectly **loop-free**



NDN: Detect & handle loops via **nonce!**

Traffic Splitting

Great benefits of using multiple paths **at the same time!**

- Load Balancing
- Gradual shifting of traffic during congestion [1, 2]
- Exploiting differences in cost and performance of paths (WiFi vs. LTE) [3]

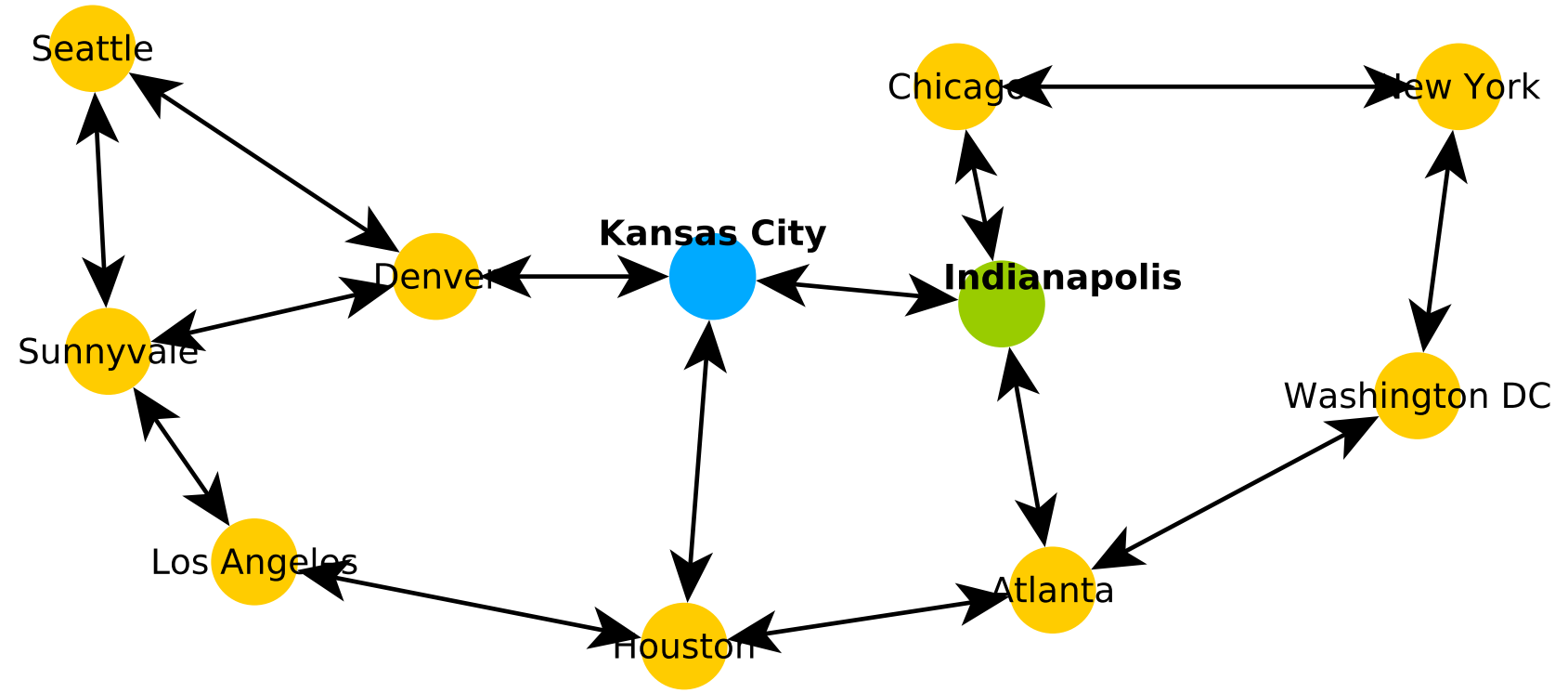
Traffic Splitting

Great benefits of using multiple paths **at the same time!**

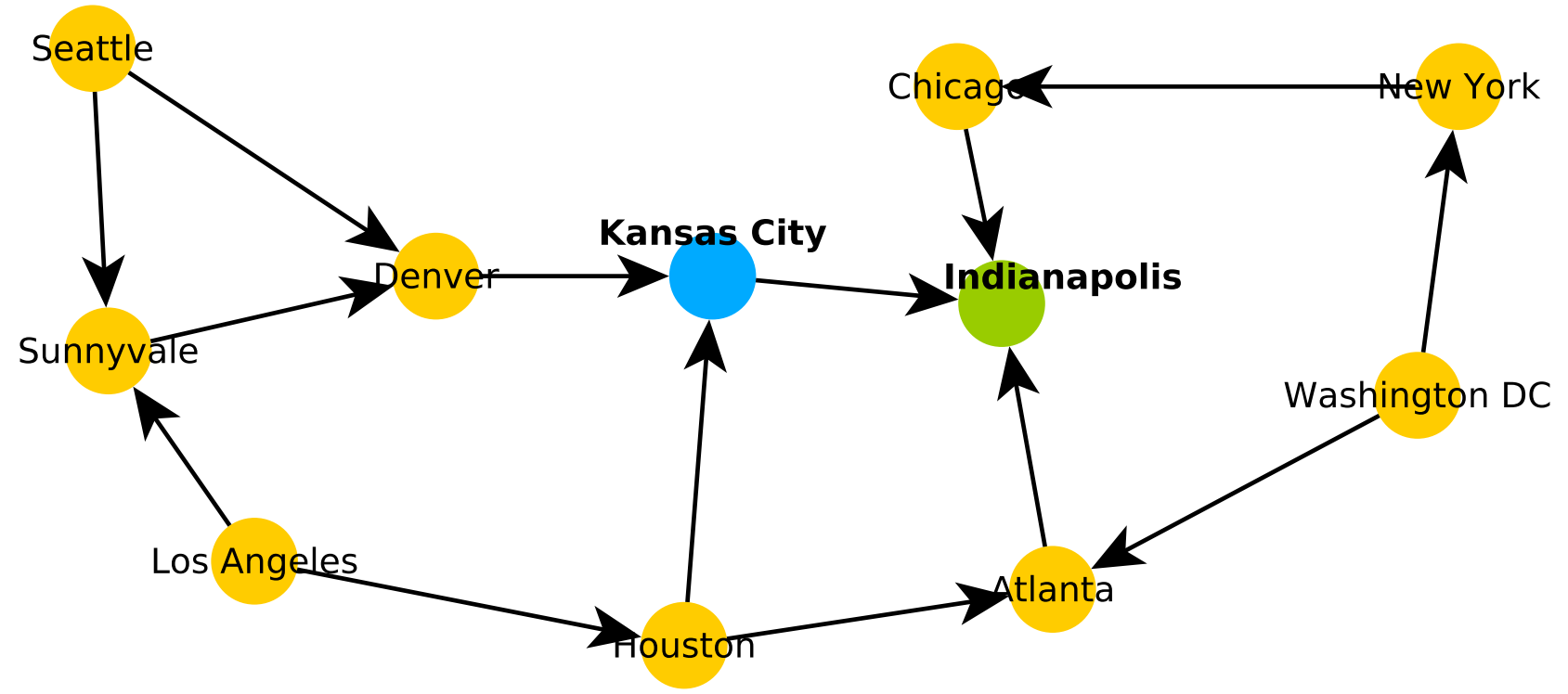
- Load Balancing
- Gradual shifting of traffic during congestion [1, 2]
- Exploiting differences in cost and performance of paths (WiFi vs. LTE) [3]

Traffic Splitting requires **Loop-Freeness** and **Path Choice**

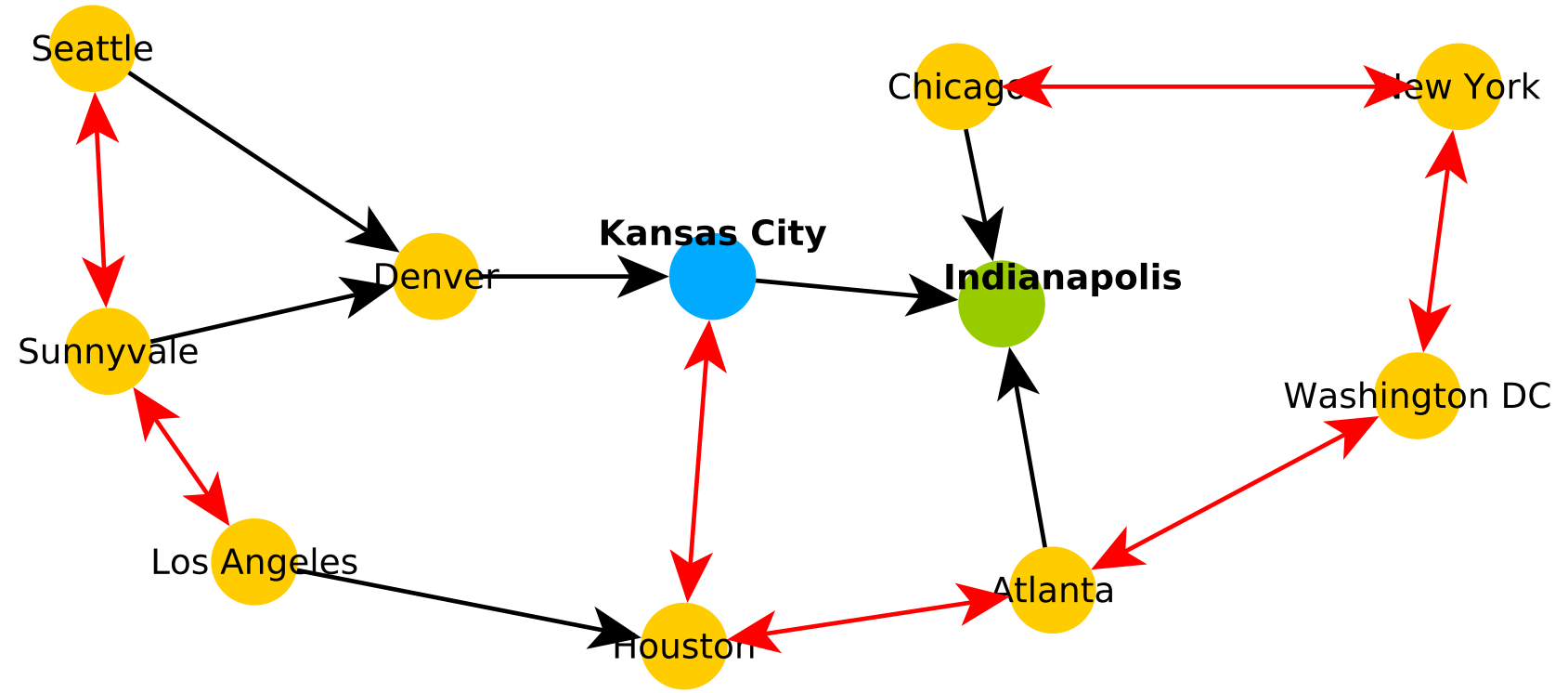
NLSR: High Path Choice, but many Loops



Loop-Free Routing: Limited Path Choice



Incoming Interface-Exclusion: Higher Path Choice



How To Achieve That?

Via Routing? **Too complex!**

1. Trad: Undirected graph $\Rightarrow$ Directed Acyclic Graph (DAG)
2. Now: $\Rightarrow$ Directed graph **without cycles longer than one hop!**

How To Achieve That?

Via Routing? **Too complex!**

1. Trad: Undirected graph $\Rightarrow$ Directed Acyclic Graph (DAG)
2. Now: $\Rightarrow$ Directed graph **without cycles longer than one hop!**

Solution:

1. **Almost Loop-free Routing** (ALR)
2. On-demand **Loop Removal** at forwarding layer

Almost Loop-Free Routing

Give each nexthop in FIB a Type: { `Downward`, `Upward`, `Disabled` }.

1. Fill FIB with Downward Nexthops
2. If prefix still has **only one nexthop:**
   - Go through all disabled nexthops:
     - If NH passes **Heuristic Loop Checks:**
       Add as Uphill; break;

Almost Loop-Free Routing

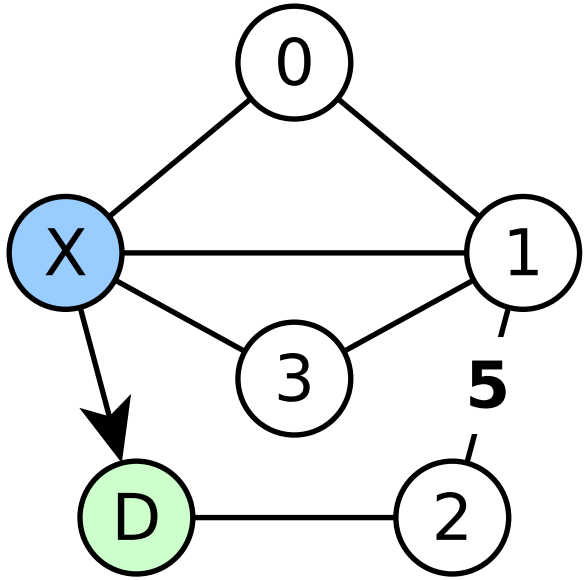
Give each nexthop in FIB a Type: { `Downward`, `Upward`, `Disabled` }.

1. Fill FIB with Downward Nexthops
2. If prefix still has **only one nexthop:**
   - Go through all disabled nexthops:
     - If NH passes **Heuristic Loop Checks:**
       Add as Uphill; break;

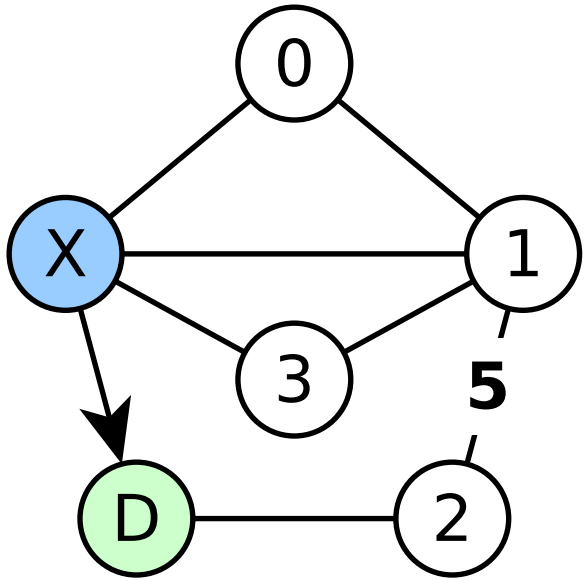
⇒ Much more nodes with **at least 2 nexthops**, and low looping chance.

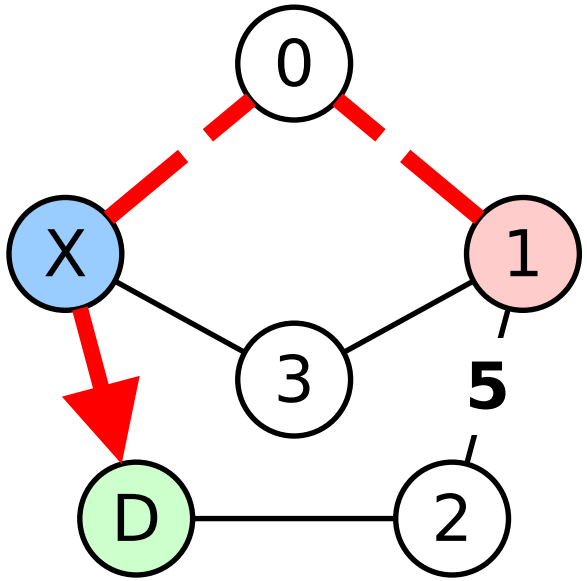
Forwarding loop: **Uphill Nexthop Removal** (UNR)

Example of Loop-Freeness Heuristic

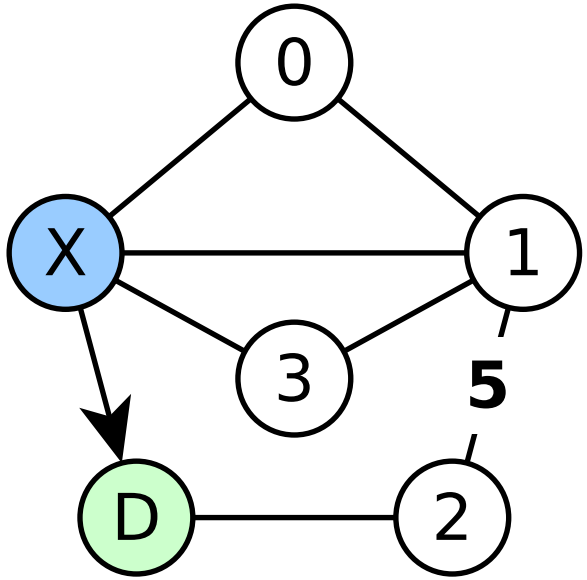


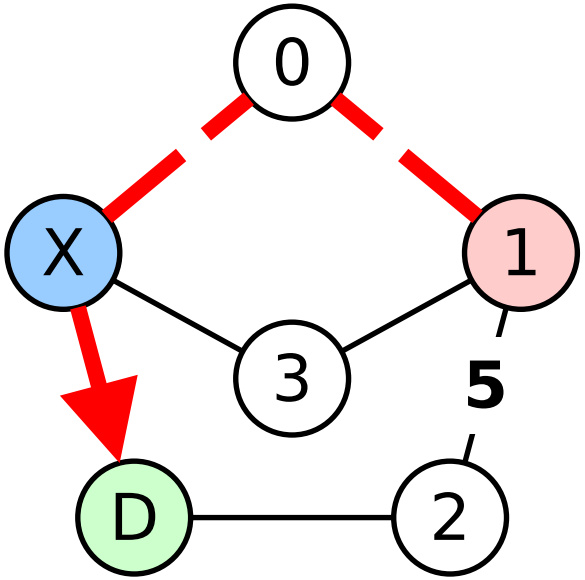
Example of Loop-Freeness Heuristic

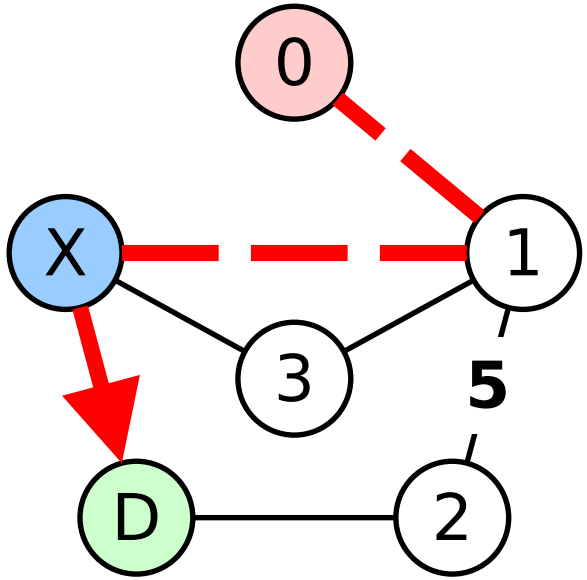




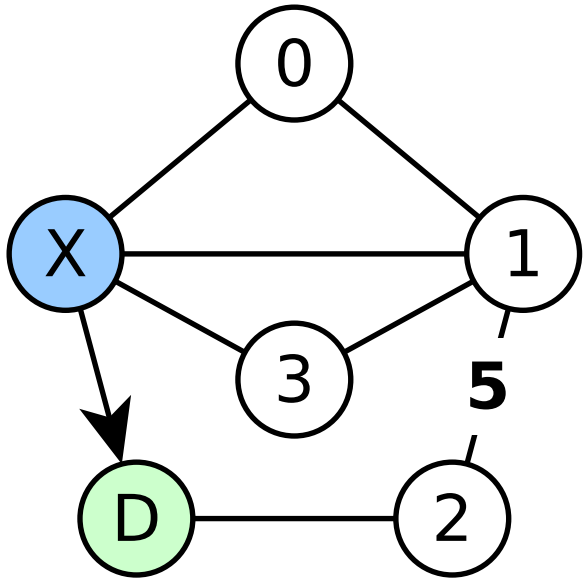
Example of Loop-Freeness Heuristic

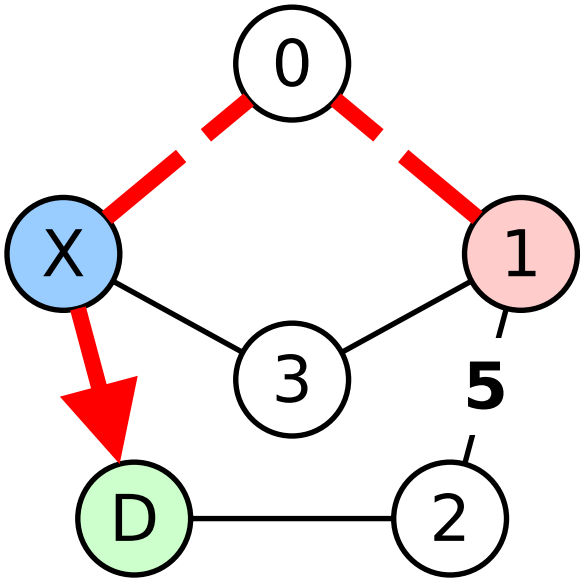


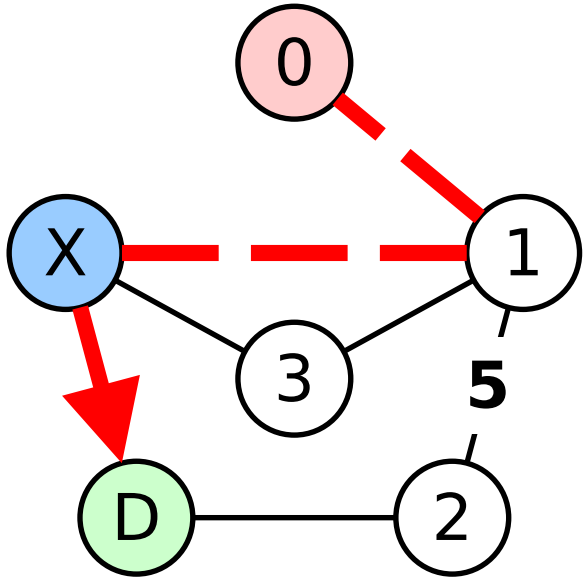


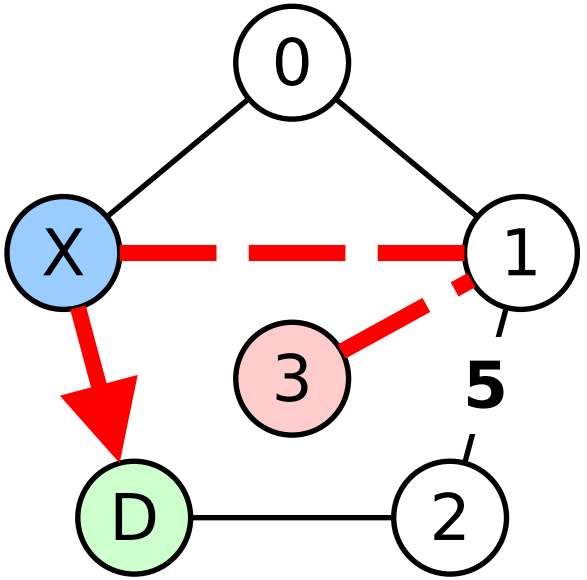


Example of Loop-Freeness Heuristic









Evaluation

Results of 8 other routing protocols in 9 topologies:

- ALR: Low computational complexity, similar to NLSR.
- UNR: Very low forwarding complexity (only 0.4% to 1.6% of FIB entries need to be changed)

Together: Completely loop-free network with **higher path choice** than loop-free routing for all tested topologies.

Summary

How to Establish Loop-Free Multipath Routes in NDN?

- **Loop-free routing** vs. Loop handling at **Forw. Layer**

Summary

How to Establish Loop-Free Multipath Routes in NDN?

- **Loop-free routing** vs. Loop handling at **Forw. Layer**

Traffic Splitting needs **Loop-freeness** and high **Path Choice**

- Loop-free routing provides former, but not latter.
- Current NDN routing vice versa.

$\Rightarrow$ Achieving both by combining **Almost-Loopfree Routing** with **Loop-removal** at the forwarding layer.

Summary

How to Establish Loop-Free Multipath Routes in NDN?

- **Loop-free routing** vs. Loop handling at **Forw. Layer**

Traffic Splitting needs **Loop-freeness** and high **Path Choice**

- Loop-free routing provides former, but not latter.
- Current NDN routing vice versa.

$\Rightarrow$ Achieving both by combining **Almost-Loopfree Routing** with **Loop-removal** at the forwarding layer.

Future Work:

- Tech Report **coming soon!**

The End

Thank you for your attention!

Questions?

Klaus Schneider
klaus@cs.arizona.edu

References I

[1] Giovanna Carofiglio, Massimo Gallo, Luca Muscariello, Michele Papalini, and Sen Wang.
Optimal multipath congestion control and request forwarding in information-centric networks.
In *ICNP*, 2013.

[2] Klaus Schneider, Cheng Yi, Beichuan Zhang, and Lixia Zhang.
A practical congestion control scheme for named data networking.
In *Proceedings of ICN 2016*, pages 21–30. ACM, 2016.

[3] Klaus M Schneider and Udo R Krieger.
Beyond network selection: Exploiting access network heterogeneity with named data networking.
In *Proceedings of the 2nd International Conference on Information-Centric Networking*, pages 137–146. ACM, 2015.