Reliable routing

Brighten Godfrey
CS 538 March 13, 2017

Paxson'96 key points

The Internet is messy in practice

- Transient loops
- Persistent loops
- Asymmetry
- Instability

How to look inside a black box

Looking inside a black box

End-to-end measurement from vantage points combined with careful statistics

A standard for the field

- End-to-End Effects of Internet Path Selection [Savage '99]
- RON [Anderson '01]
- Related area: network tomography

Many resources now available

- PlanetLab, Seattle P2P testbed, RouteViews, DIMES, CAIDA, ...

Name	Description
adv	Advanced Network & Services, Armonk, NY
austr	University of Melbourne, Australia
austr2	University of Newcastle, Australia
batman	National Center for Atmospheric Research, Boulder, CO
bnl	Brookhaven National Lab, NY
bsdi	Berkeley Software Design, Colorado Springs, CO
connix	Caravela Software, Middlefield, CT
harv	Harvard University, Cambridge, MA
inria	INRIA, Sophia, France
korea	Pohang Institute of Science and Technology, South Korea
lbl	Lawrence Berkeley Lab, CA
lbli	LBL computer connected via ISDN, CA
mid	MIDnet, Lincoln, NE
mit	Massachusetts Institute of Technology, Cambridge, MA
ncar	National Center for Atmospheric Research, Boulder, CO
near	NEARnet, Cambridge, Massachusetts
nrao	National Radio Astronomy Observatory, Charlottesville, VA
oce	Oce-van der Grinten, Venlo, The Netherlands
panix	Public Access Networks Corporation, New York, NY
pubnix	Pix Technologies Corp., Fairfax, VA
rain	RAINet, Portland, Oregon
sandia	Sandia National Lab, Livermore, CA
sdsc	San Diego Supercomputer Center, CA
sintef1	University of Trondheim, Norway
sintef2	University of Trondheim, Norway
sri	SRI International, Menlo Park, CA
ucl	University College, London, U.K.
ucla	University of California, Los Angeles
ucol	University of Colorado, Boulder
ukc	University of Kent, Canterbury, U.K.
umann	University of Mannheim, Germany
umont	University of Montreal, Canada
unij	University of Nijmegen, The Netherlands
usc	University of Southern California, Los Angeles
ustutt	University of Stuttgart, Germany
wustl	Washington University, St. Louis, MO
xor	XOR Network Engineering, East Boulder, CO

[Paxson's vantage points]

Network reliability in context

Physical layer reliability

Performance reliability / quality of service

Congestion and capacity planning

Management issues

- *"The presence of persistent loops of durations on the order of hours is quite surprising, and suggests a lack of good tools for diagnosing network problems."* – Paxson

Basic routing reachability

Today

- Does the routing protocol get packets from *A* to *B*?

Control & data speeds don't match

Reliability problems in Internet routing

- Basic issue: controlling a distributed system => inconsistent state across routers => loops, black holes
- Also in link state, distance vector

Problem: control plane is slow...

- Control plane routing does eventually converge!
- But may take 100s of milliseconds; milliseconds possible after careful tuning of protocol timers

...and data plane is fast

- Sending 50 byte packet at 40 Gbps = 10 nanoseconds

Reliability in the data plane

Fast path (data plane) needs failure reaction!

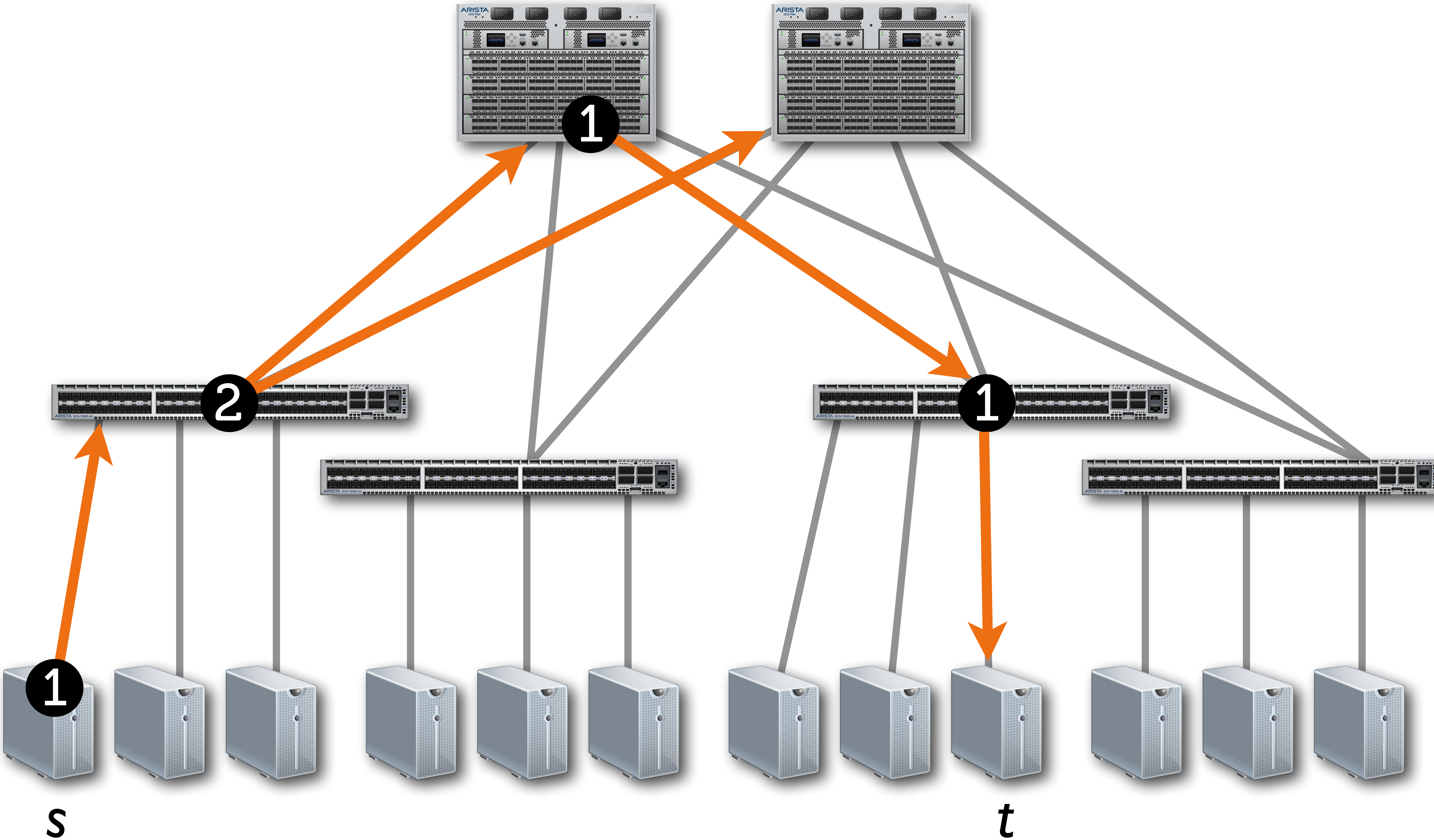
Rest of this lecture: building a solution

Techniques in practice

Equal Cost Multipath (ECMP)

- Control plane produces not one next-hop, but many
- Next hops must be closer to destination (so no loops)
- Data plane sends packet to any next-hop that's working

How many next-hops in ECMP?



Techniques in practice

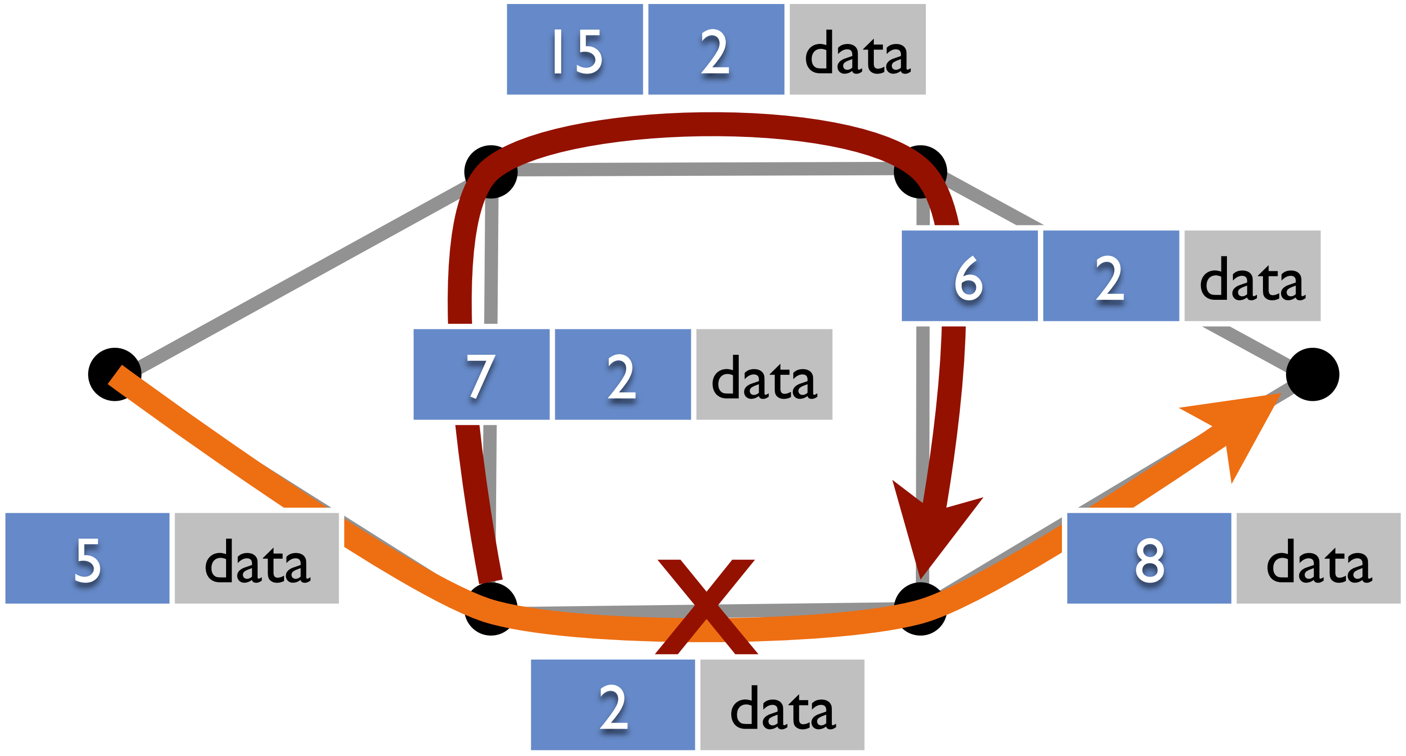
Equal Cost Multipath (ECMP)

- Control plane produces not one next-hop, but many
- Next hops must be closer to destination (so no loops)
- Data plane sends packet to any next-hop that’s working
- Defeated by even a single link failure in some cases

MPLS Fast Re-Route link protection

- Explicit backup path for each failure case (link or node failure)

MPLS FRR Link Protection



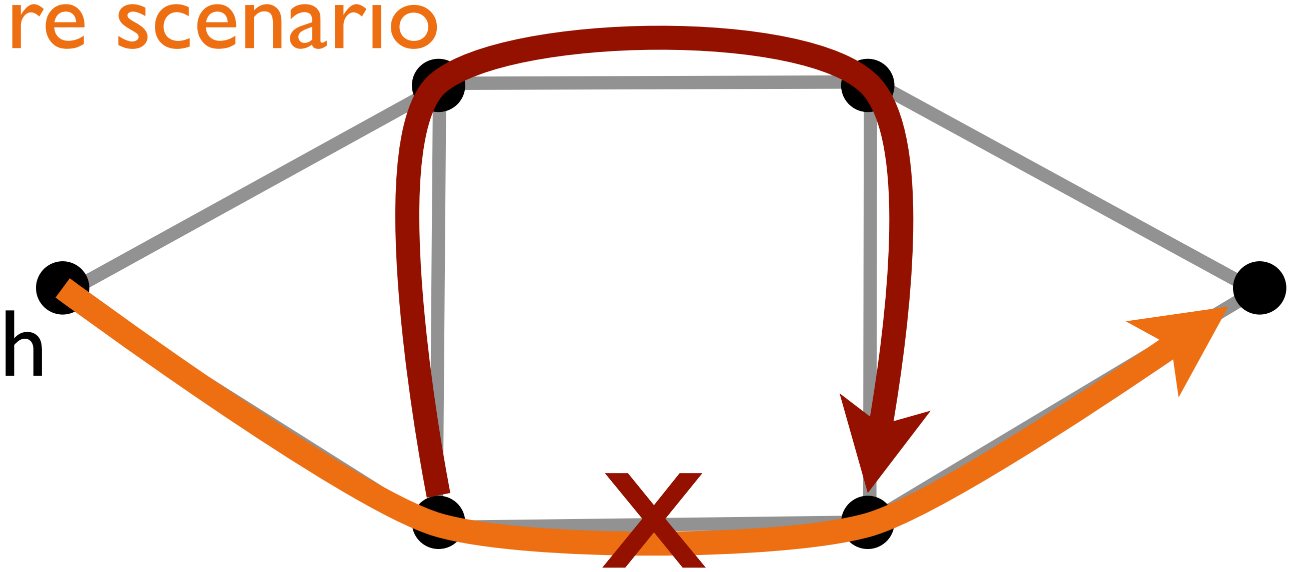
Techniques in practice

Equal Cost Multipath (ECMP)

- Control plane produces not one next-hop, but many
- Next hops must be closer to destination (so no loops)
- Data plane sends packet to any next-hop that's working
- Defeated by even a single link failure

MPLS Fast Re-Route link protection

- Explicit backup path for each link
- Protects against single failure scenario (shared risk link group)
- Uses more FIB entries
- Not shortest alternate path



DOOMED!! .... ?

Holy Grail: "Ideal connectivity"

- Data plane always correctly forwards packets towards destination, even with arbitrary link failures

Is it possible?

- Yes!
- BGP, OSPF, RIP, ISIS, ..., all have loops & black holes during convergence, ultimately causing packet loss
- But that is not fundamentally necessary!

5 minutes in small group:
Devise a correct solution

Ideal connectivity

5 minutes in small group:
Devise a correct solution

1. Every packet is eventually forwarded to destination correctly
   - Assume: arbitrary failures, but a path exists
   - Assume: no congestion or physical layer problems
2. Simple technique implementable in data plane
   - Feel free to play with packet header formats, protocols, etc.

Achieving ideal connectivity

Ideal connectivity: correct, but...

The random walk

- If failure encountered, set a "random walk" bit in packet
- Whenever packet has random walk bit, send to random neighbor
- Slightly silly solution

Failure-carrying packets (FCP)

Achieving convergence-free routing using failure-carrying packets
Lakshminarayanan, Caesar, Rangan, Anderson, Shenker, Stoica
SIGCOMM 2007

Approach

- Link state routing + link failure info carried inside packet
- Router recomputes shortest paths on the fly given new information inside packet

Key points

- Separate two functions: long-term topology distribution, handling transient changes
- Trick: carry topology updates in packet
- Demonstrates feasibility of ideal connectivity

Failure-carrying packets (FCP)

Achieving convergence-free routing using failure-carrying packets
Lakshminarayanan, Caesar, Rangan, Anderson, Shenker, Stoica
SIGCOMM 2007

Approach

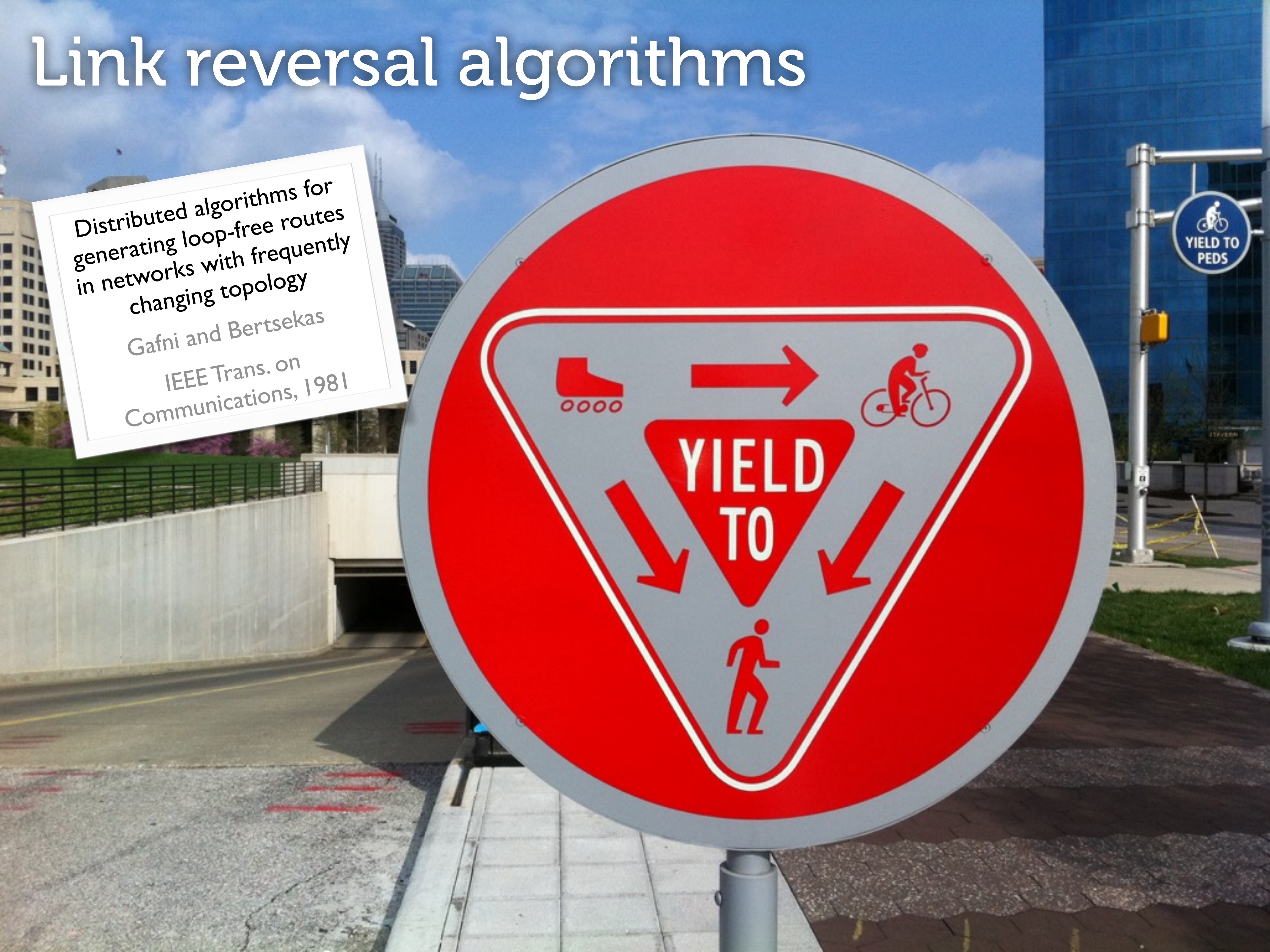
- Link state routing + link failure info carried inside packet
- Router recomputes shortest paths on the fly given new information inside packet

Difficult for data plane.
Can we do better?

Key points

- Separate two functions: long-term topology distribution, handling transient changes
- Trick: carry topology updates in packet
- Demonstrates feasibility of ideal connectivity

Link reversal algorithms



Distributed algorithms for generating loop-free routes in networks with frequently changing topology

Gafni and Bertsekas

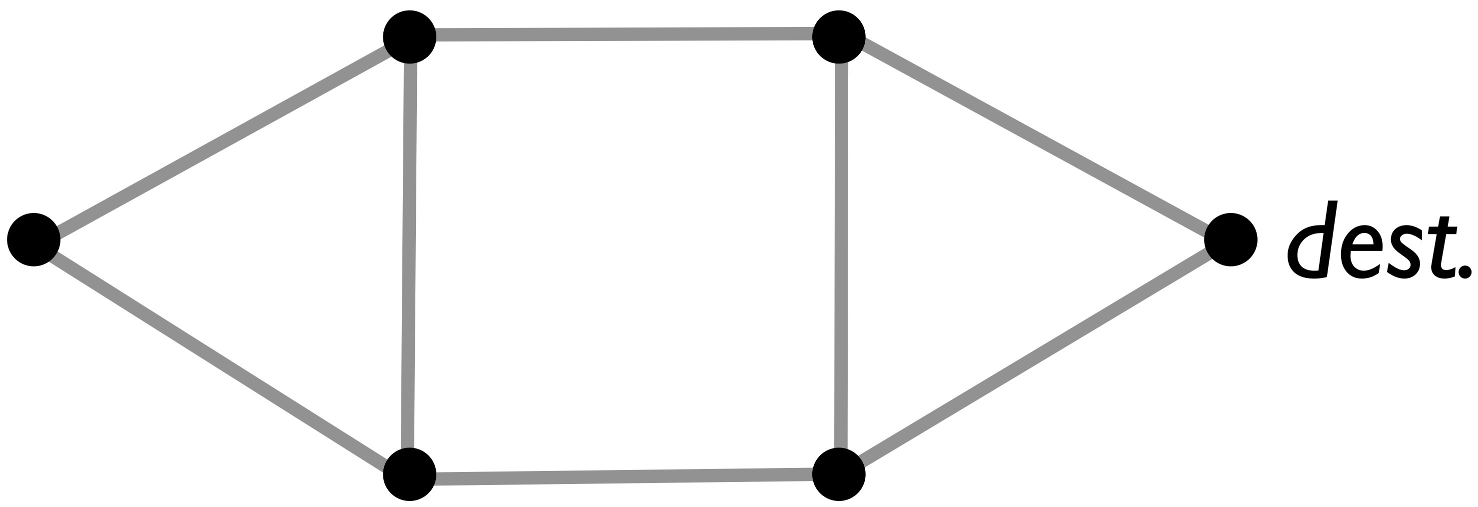
IEEE Trans. on Communications, 1981

Link reversal algorithms

Distributed algorithms for generating loop-free routes in networks with frequently changing topology

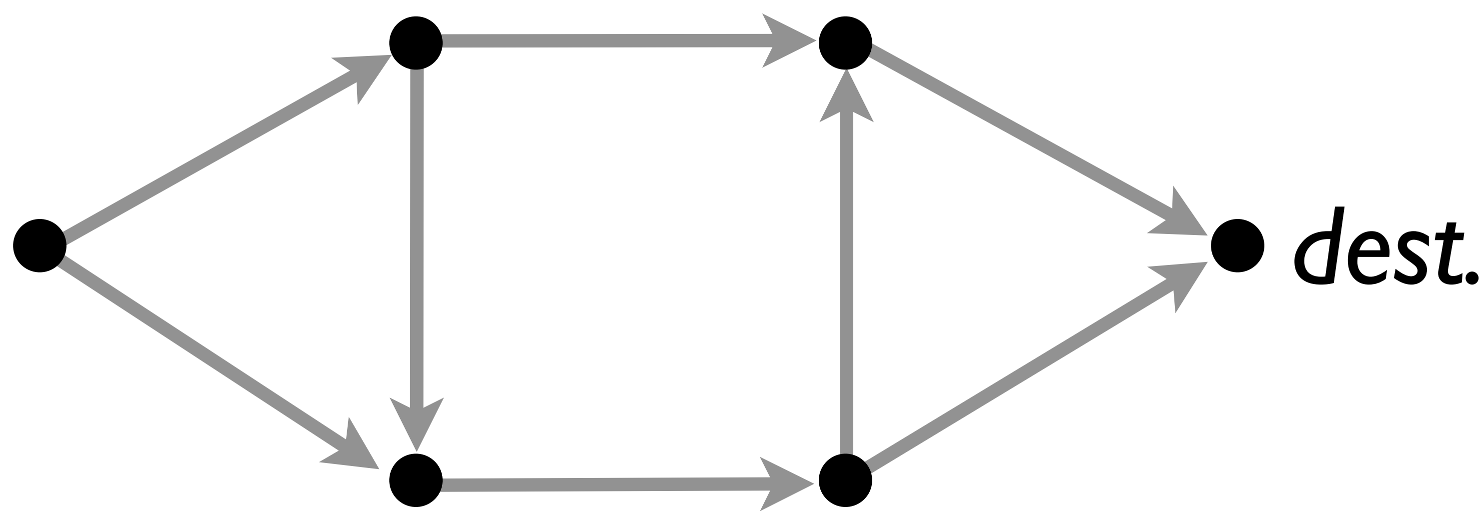
Gafni and Bertsekas

IEEE Trans. on Communications, 1981



Link reversal algorithms

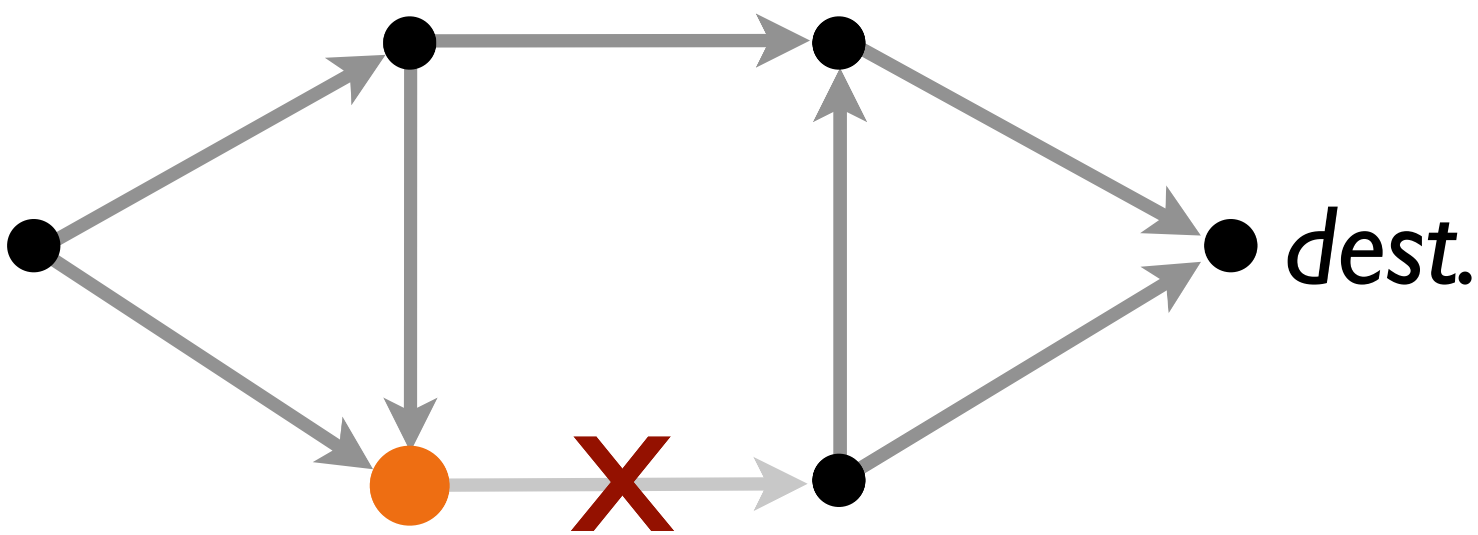
For each destination: begin with a directed acyclic graph (DAG) where destination is the sole sink



Link reversal algorithms

At each node:

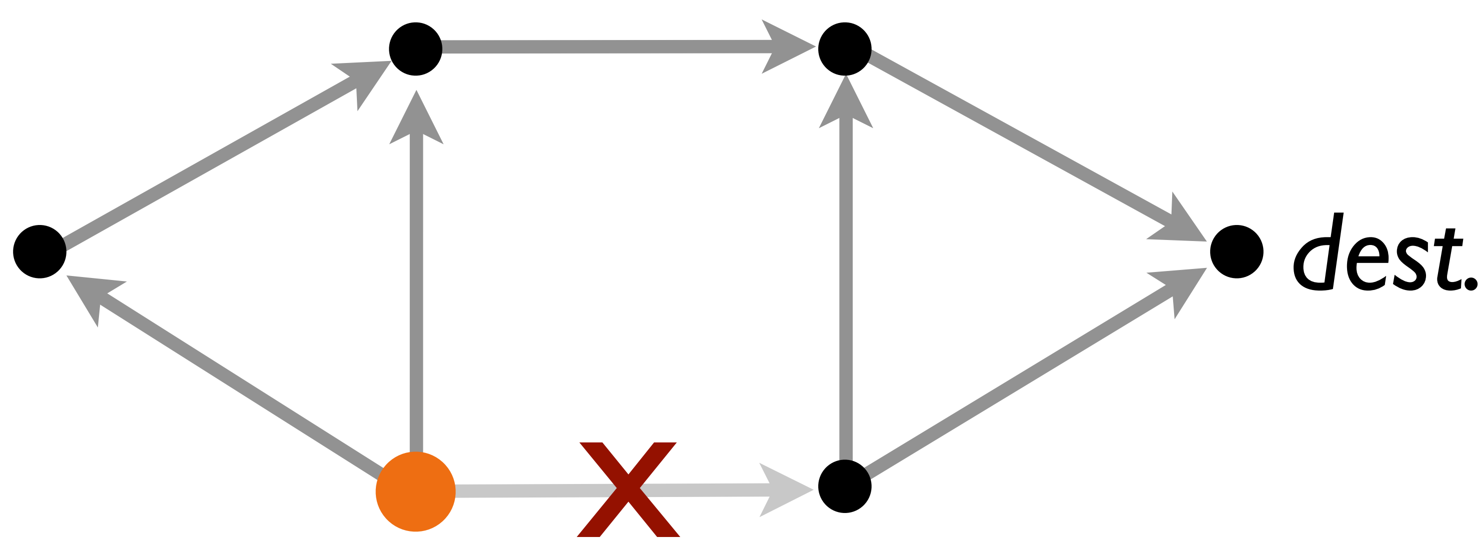
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

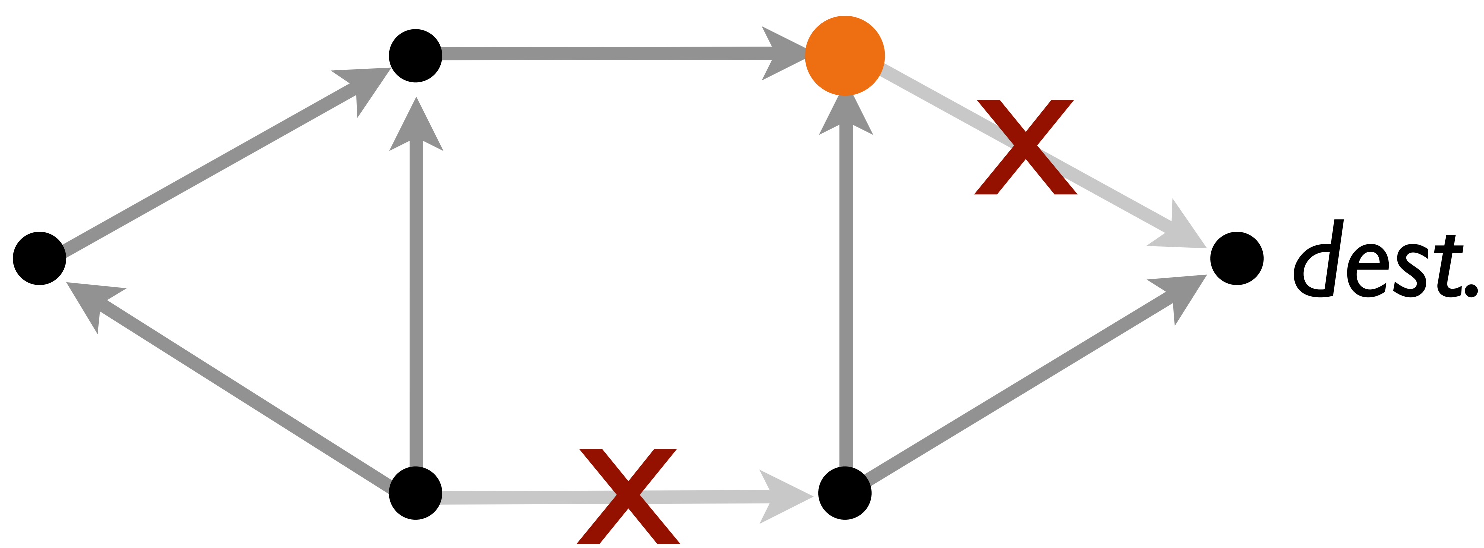
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

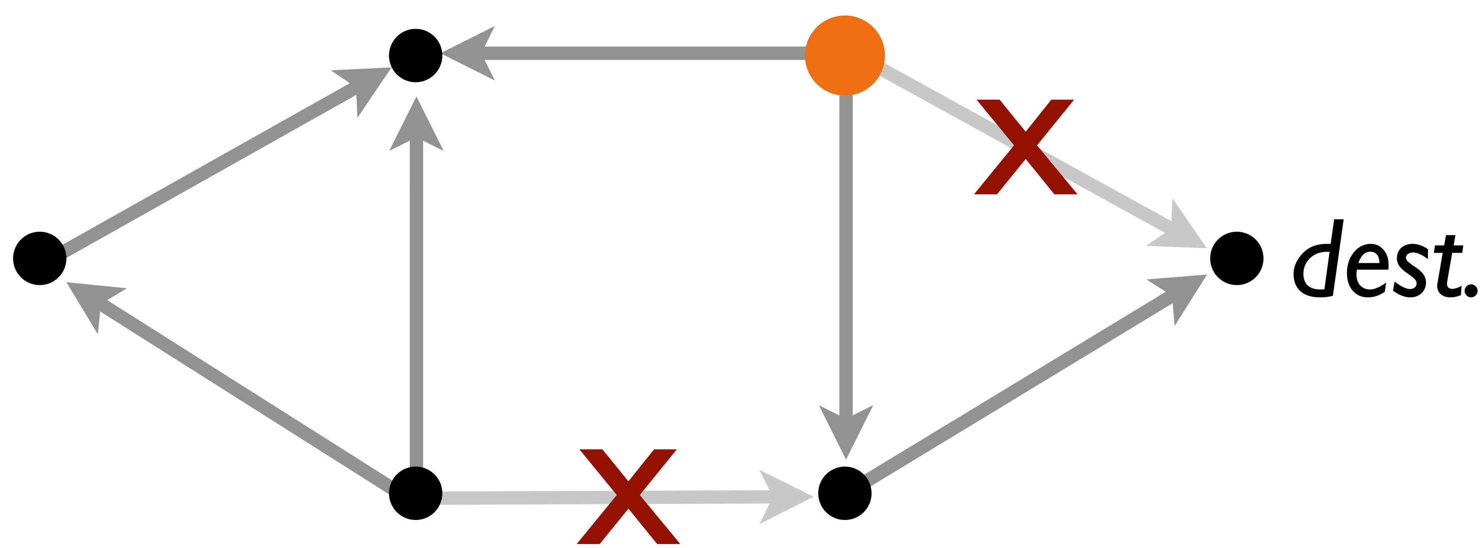
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

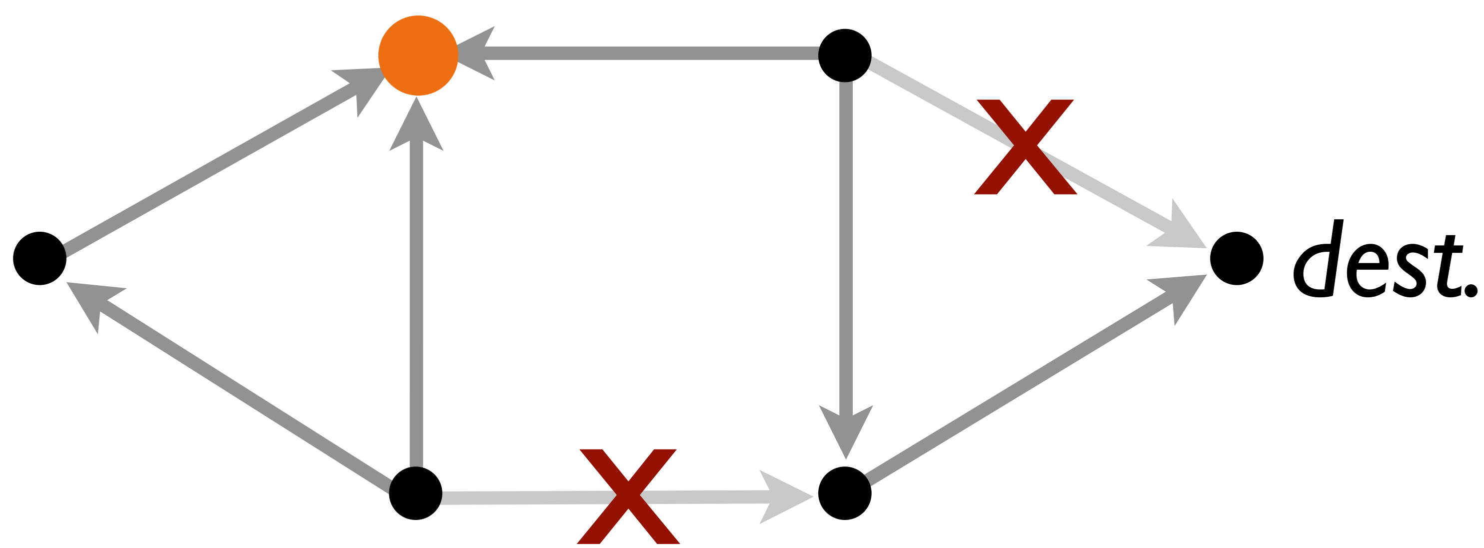
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

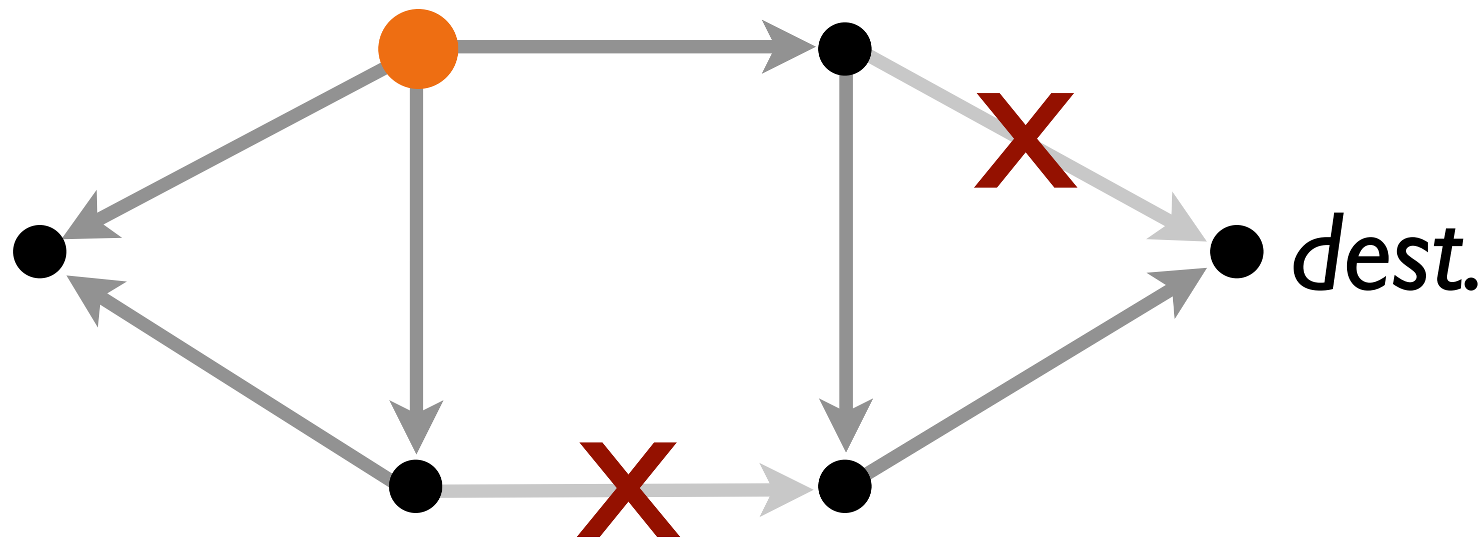
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

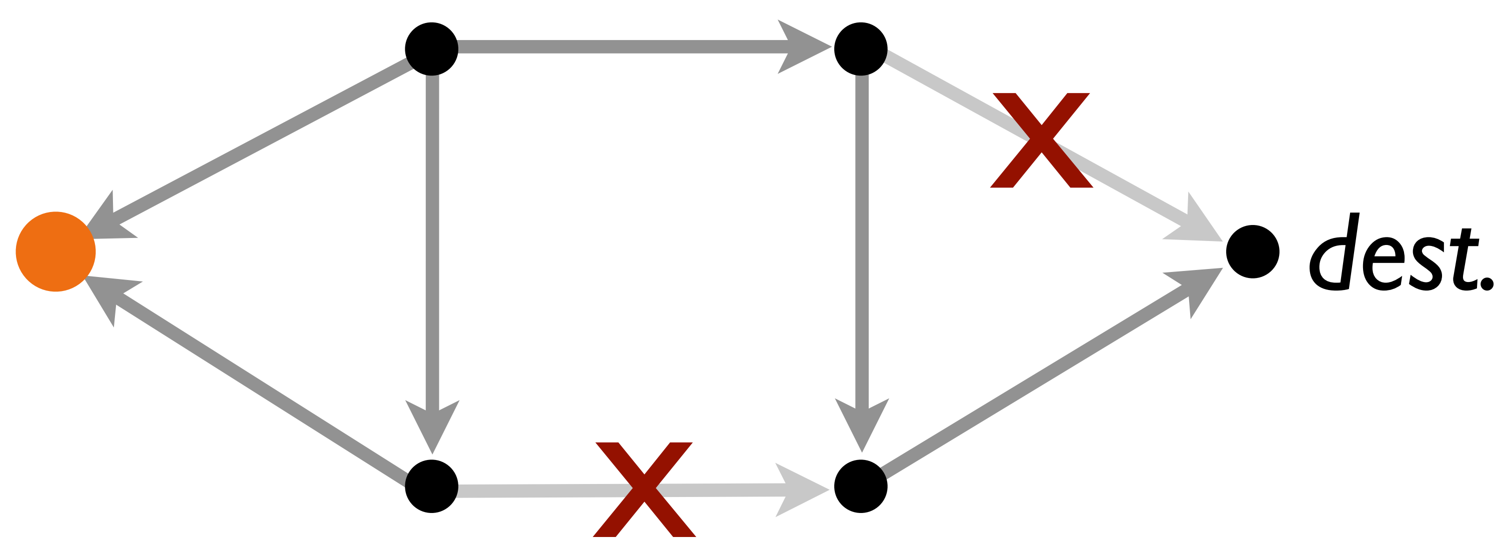
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

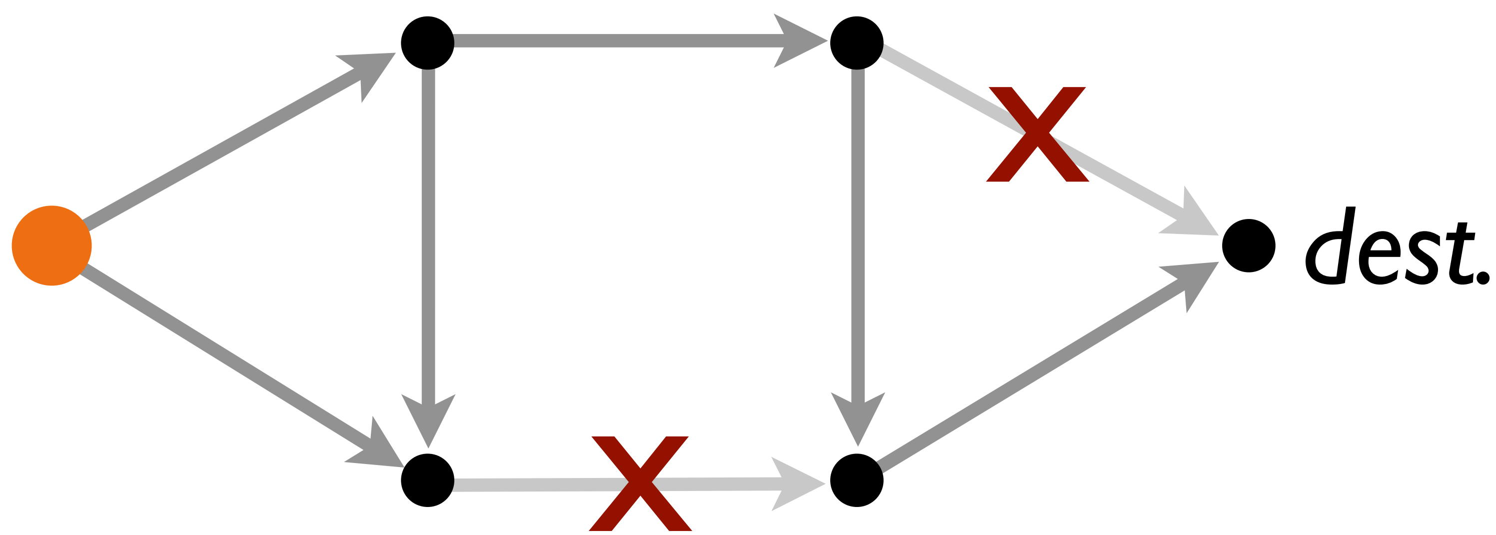
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

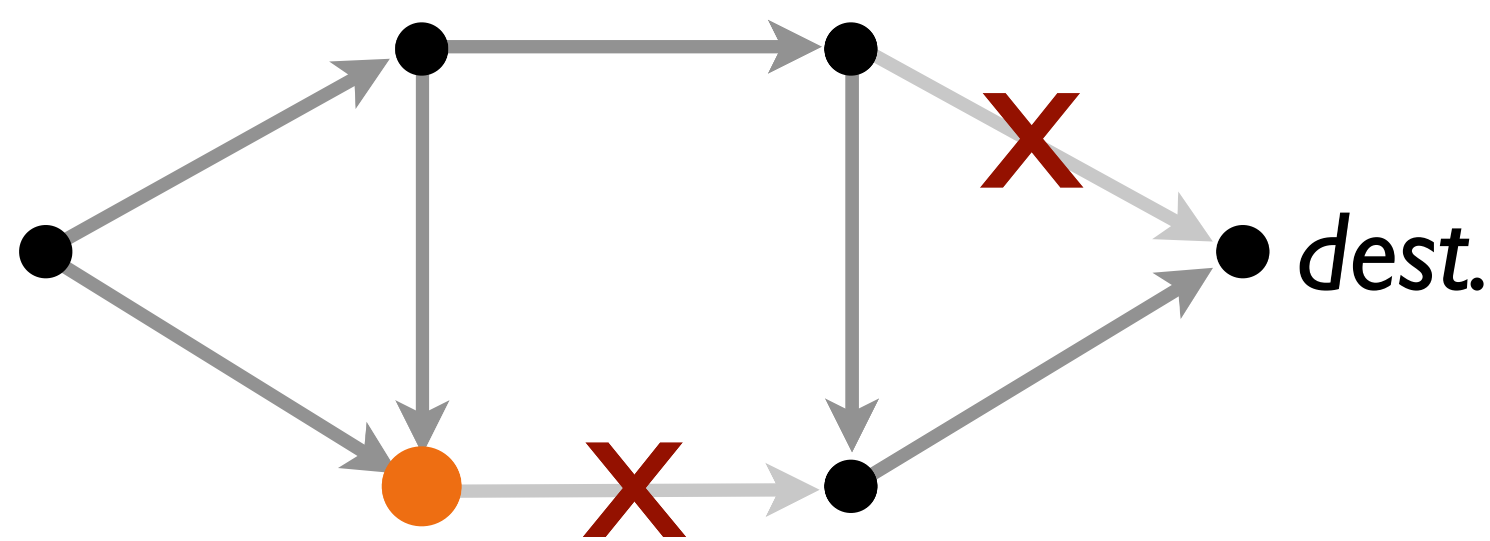
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

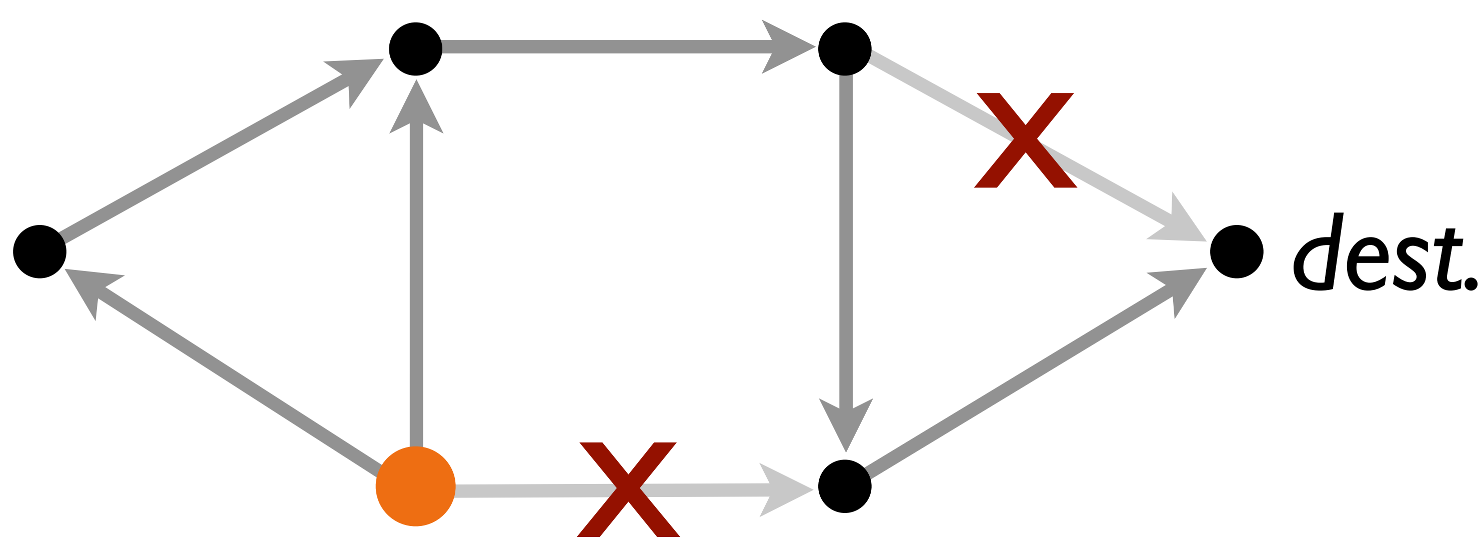
- If ever all links point inward,
  - Reverse all links



Link reversal algorithms

At each node:

- If ever all links point inward,
  - Reverse all links



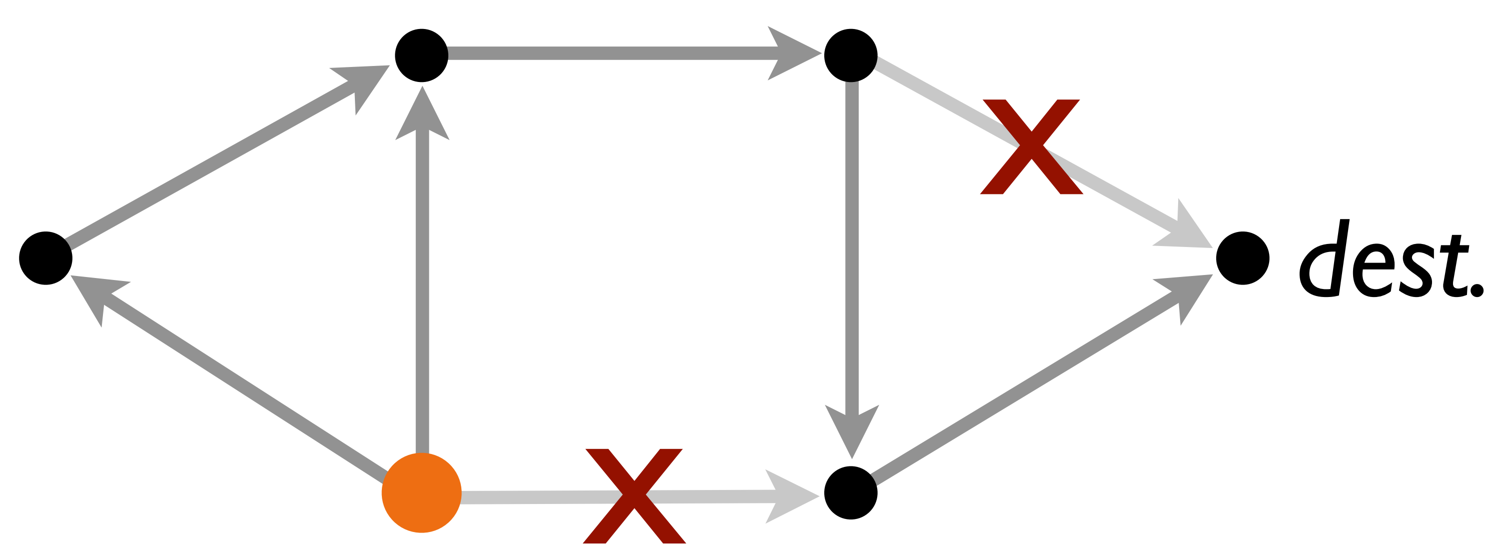
Whew! Done!

In the end, only one link flipped!

Guaranteed to converge

Define stable node: no more reversals

- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all

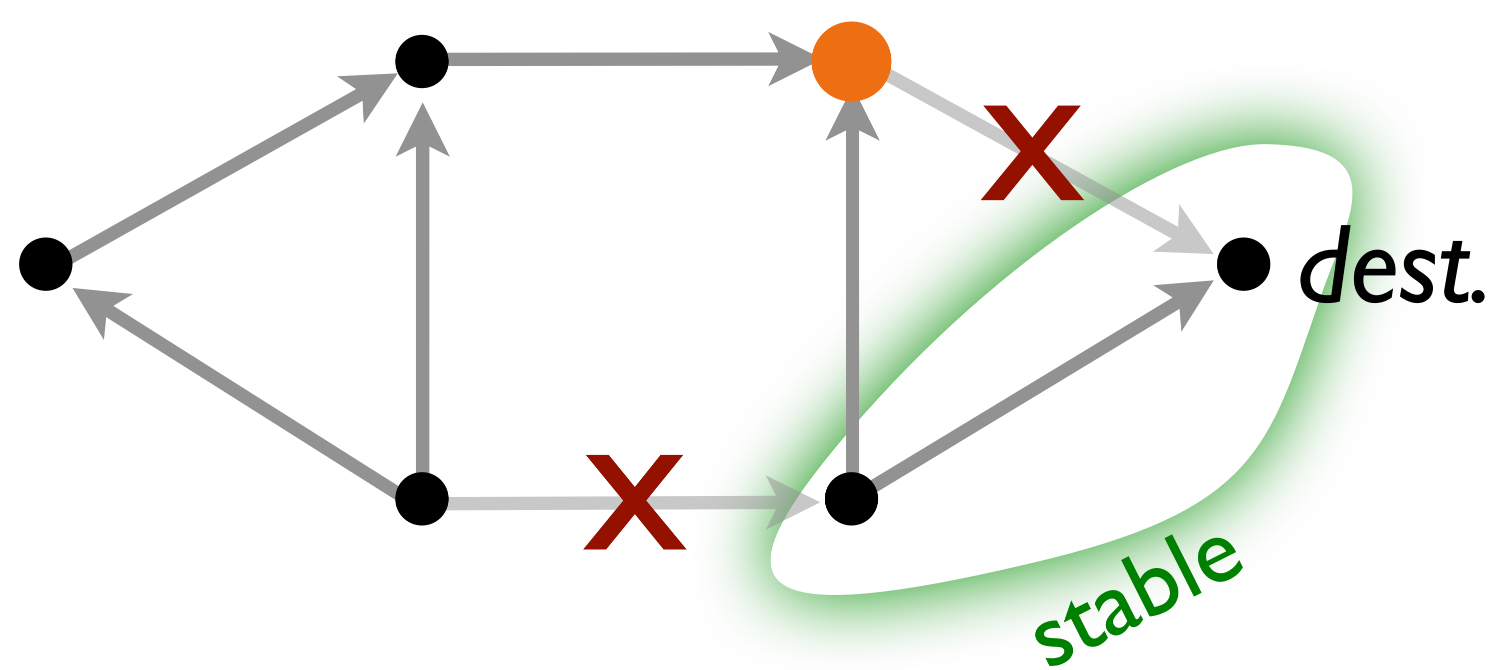


Let's return to the beginning before convergence...

Guaranteed to converge

Define stable node: no more reversals

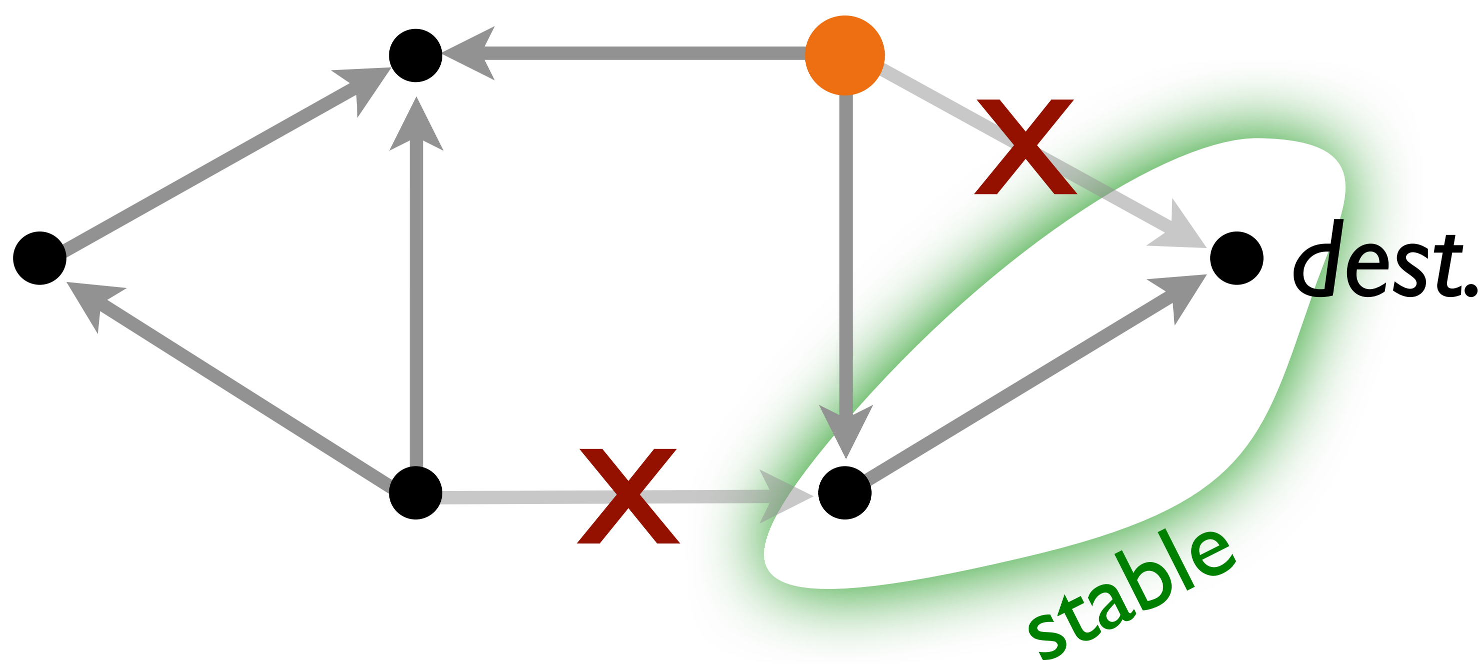
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

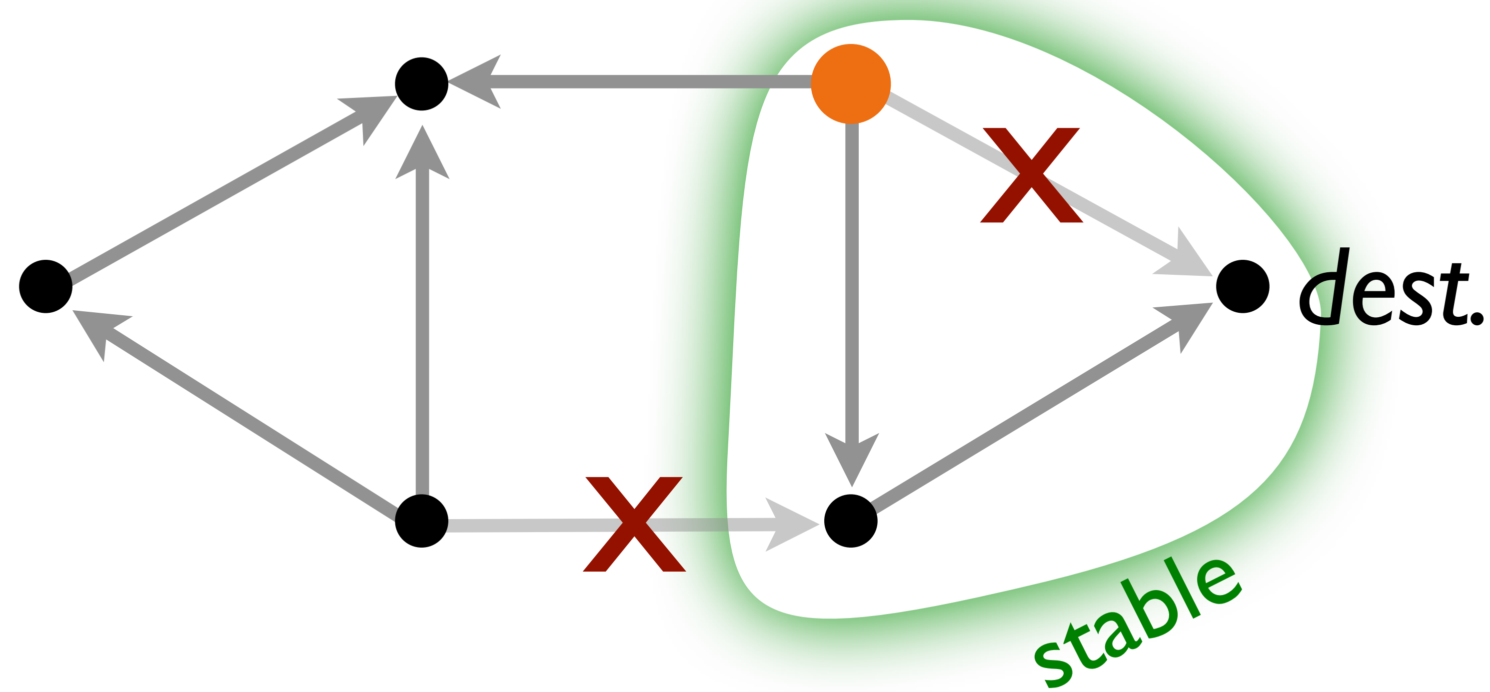
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

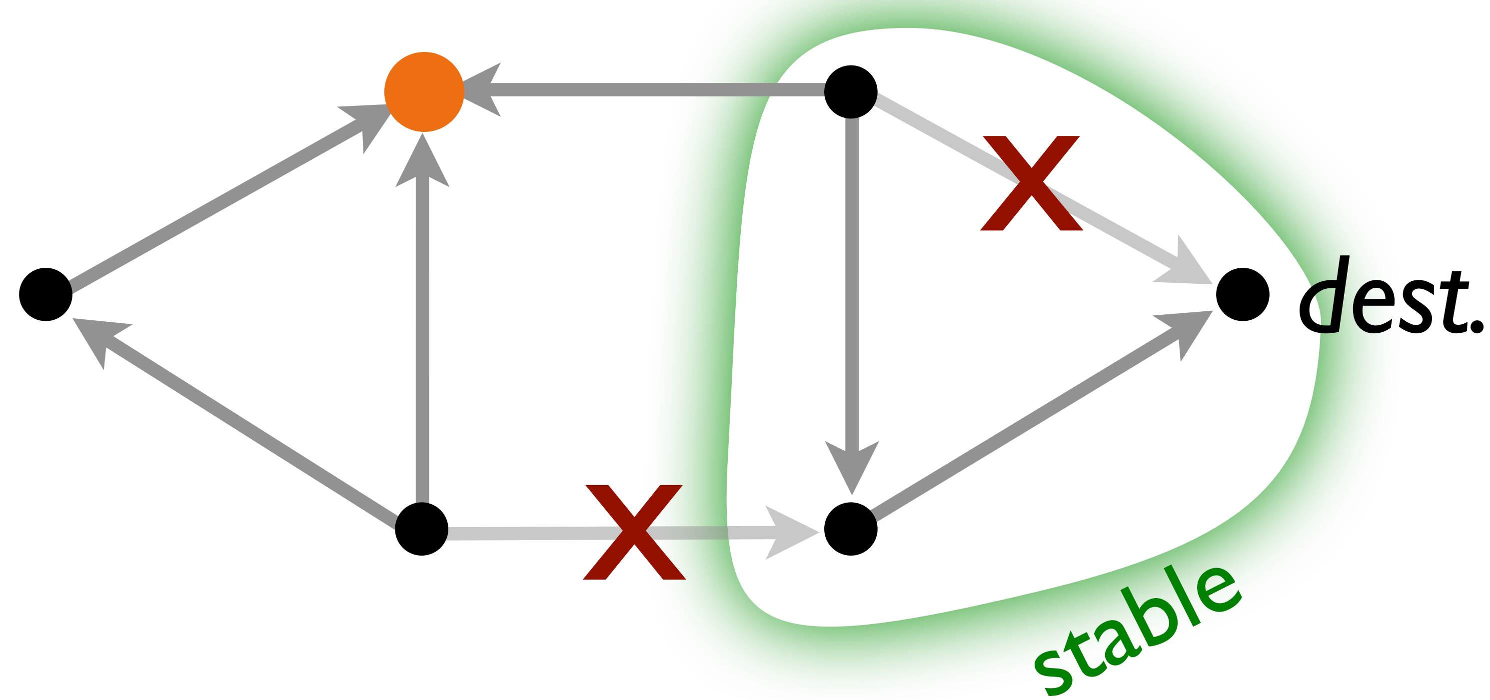
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

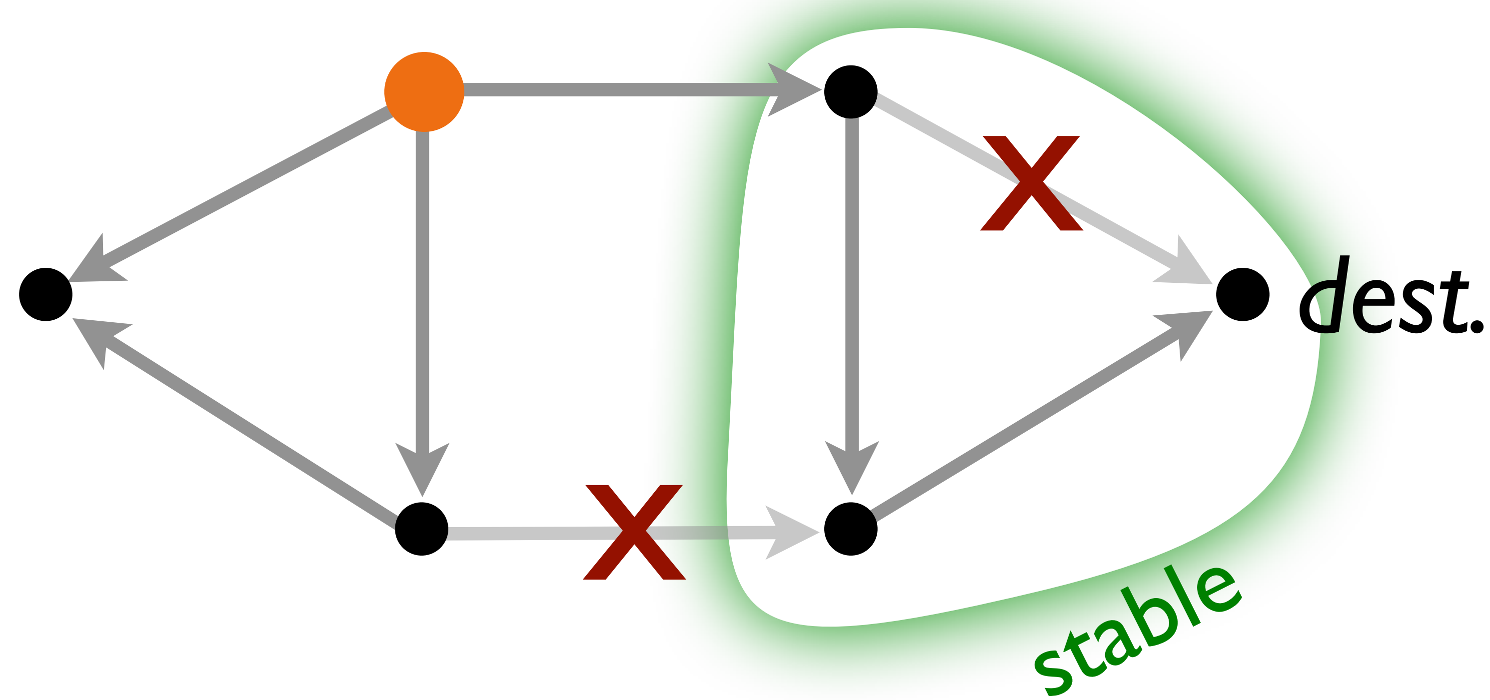
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

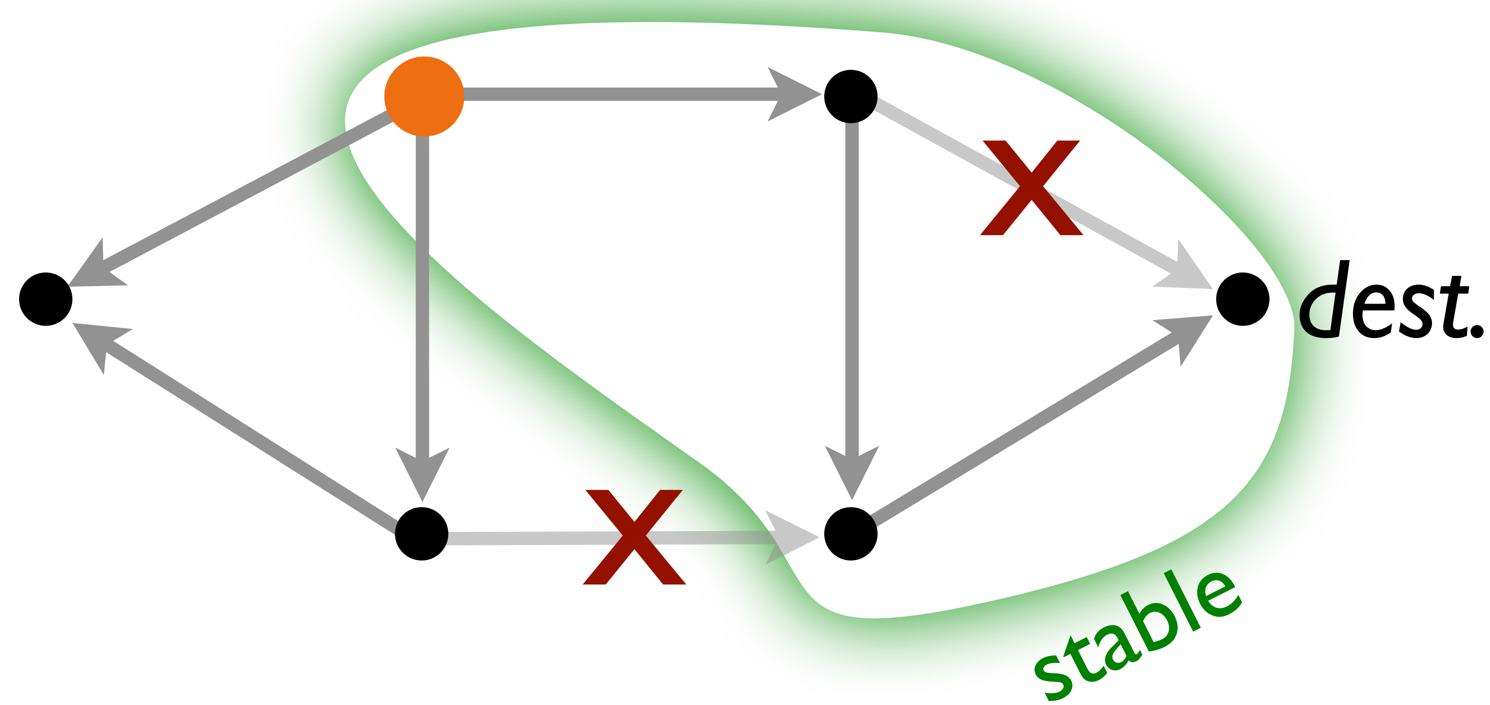
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

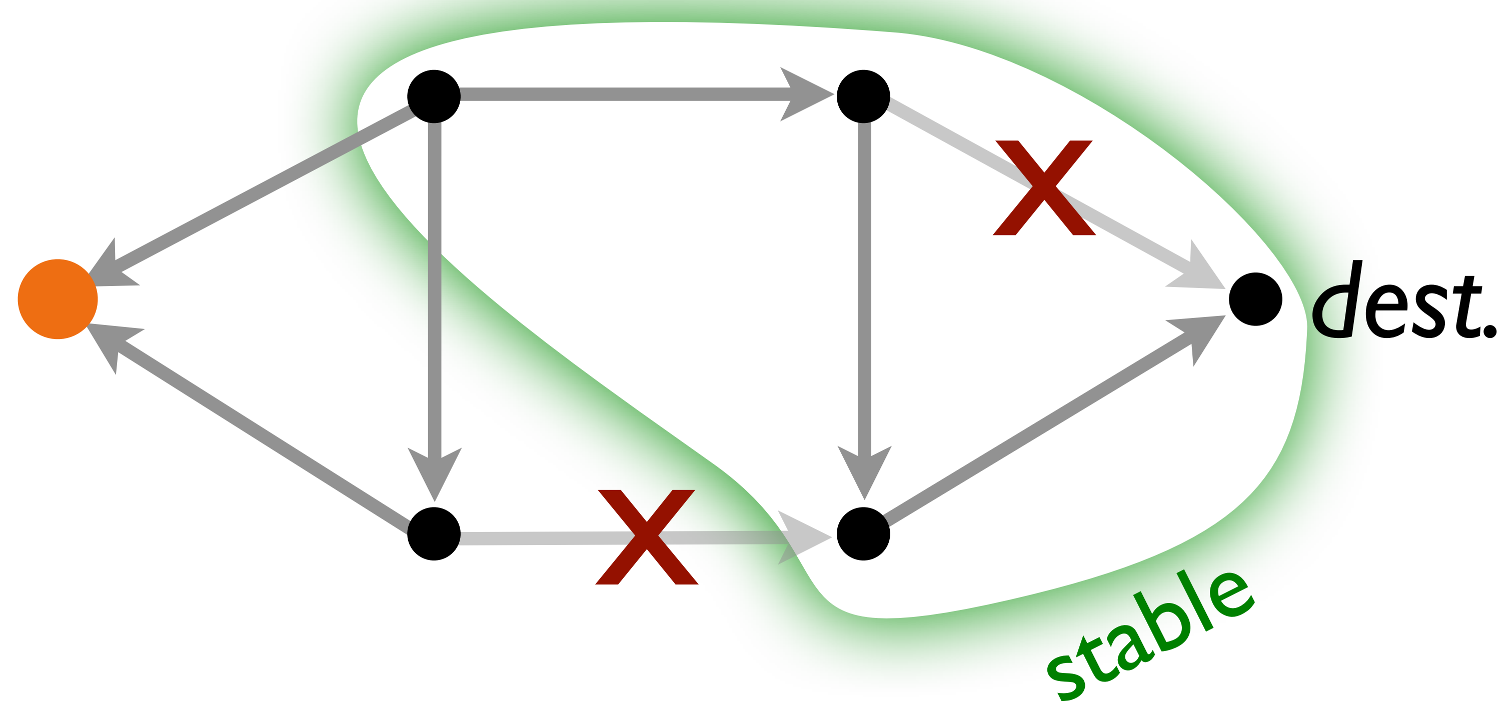
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

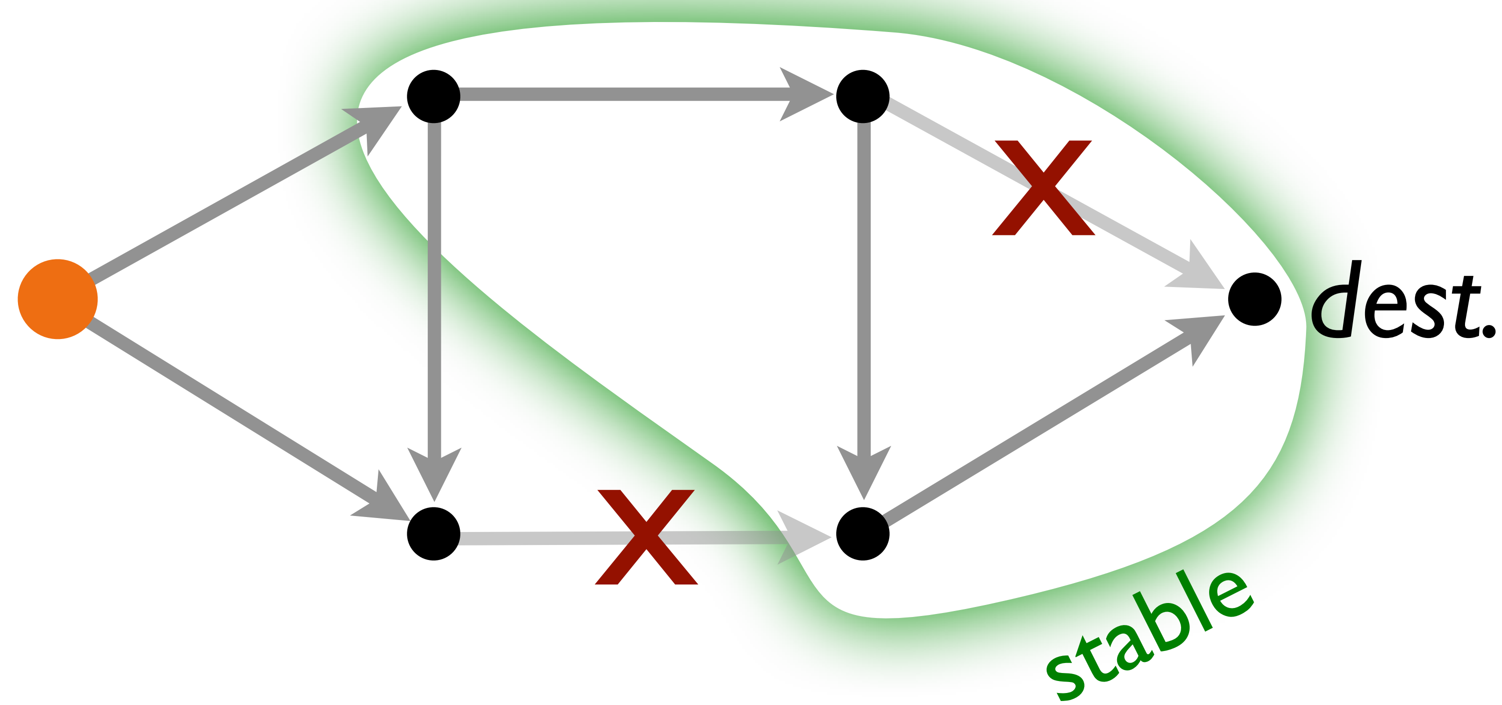
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

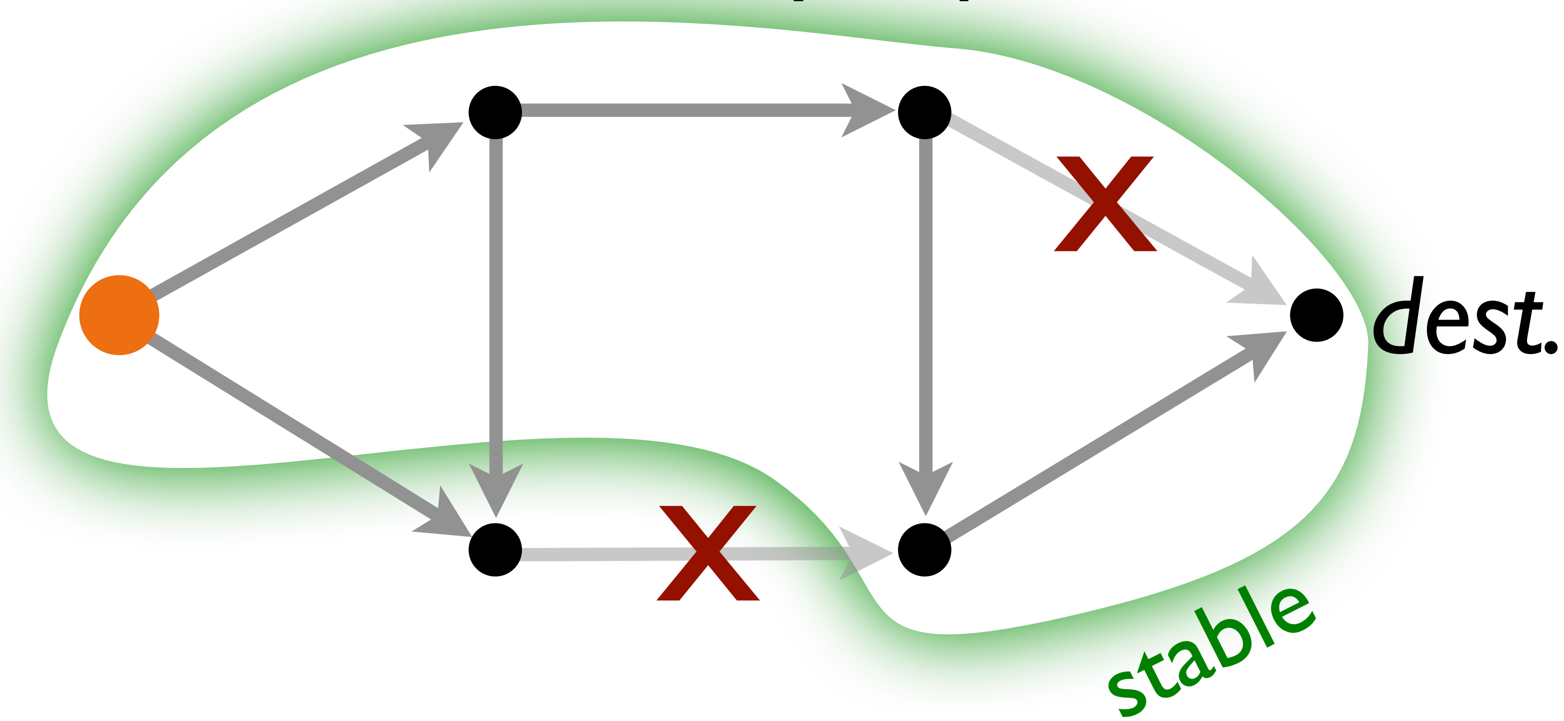
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

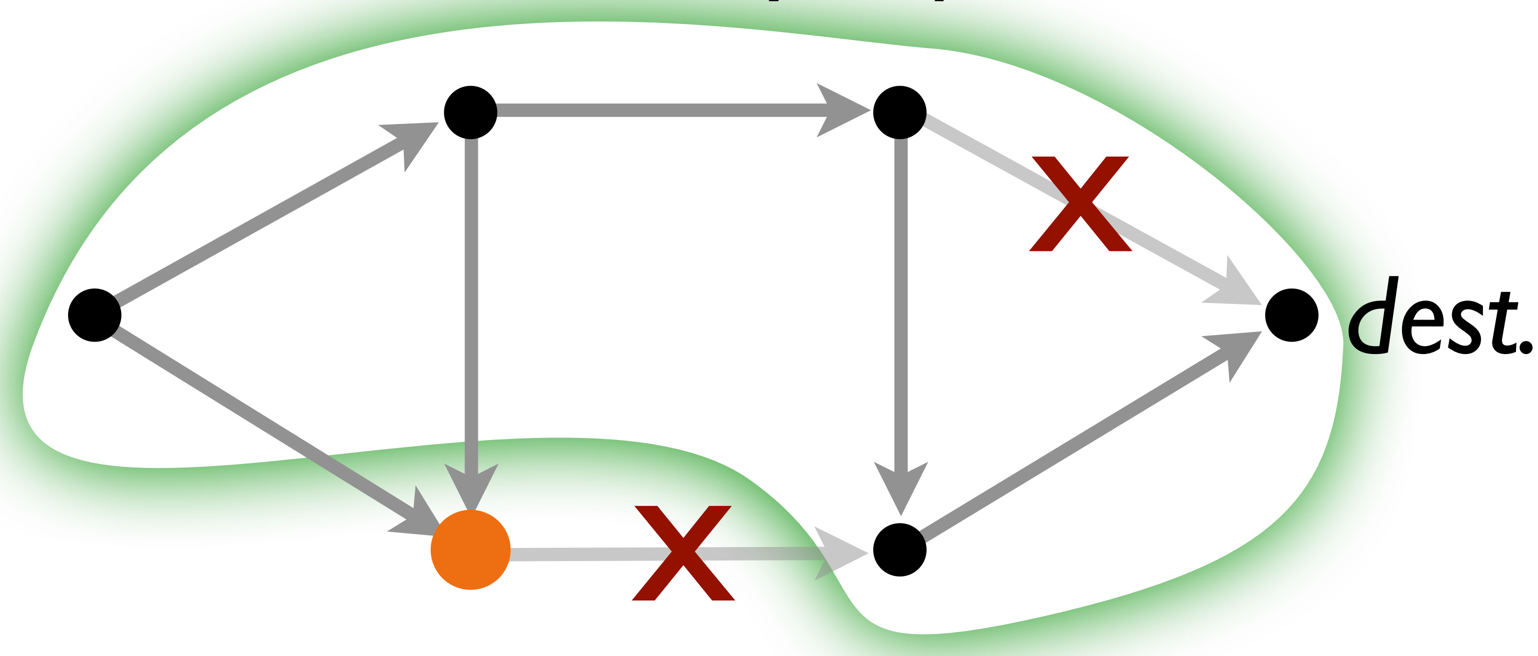
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

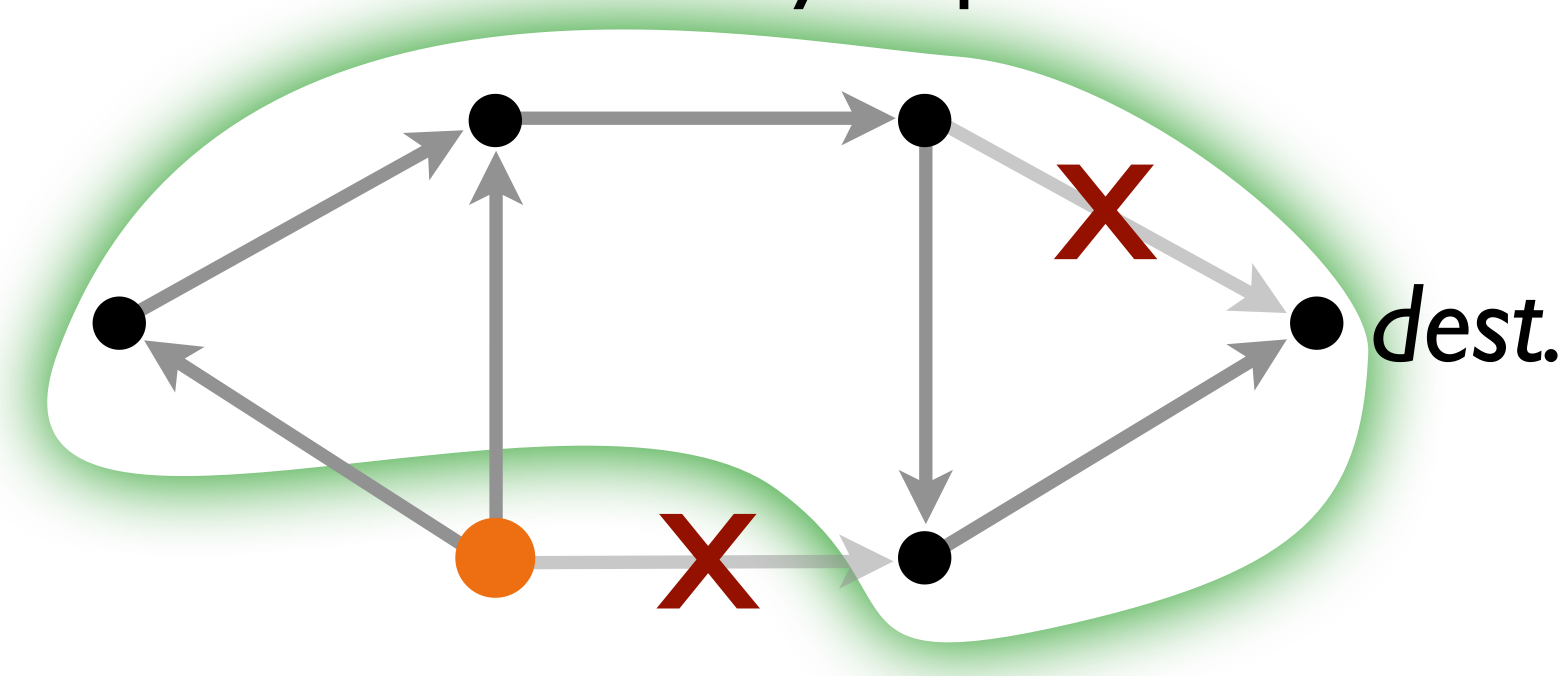
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

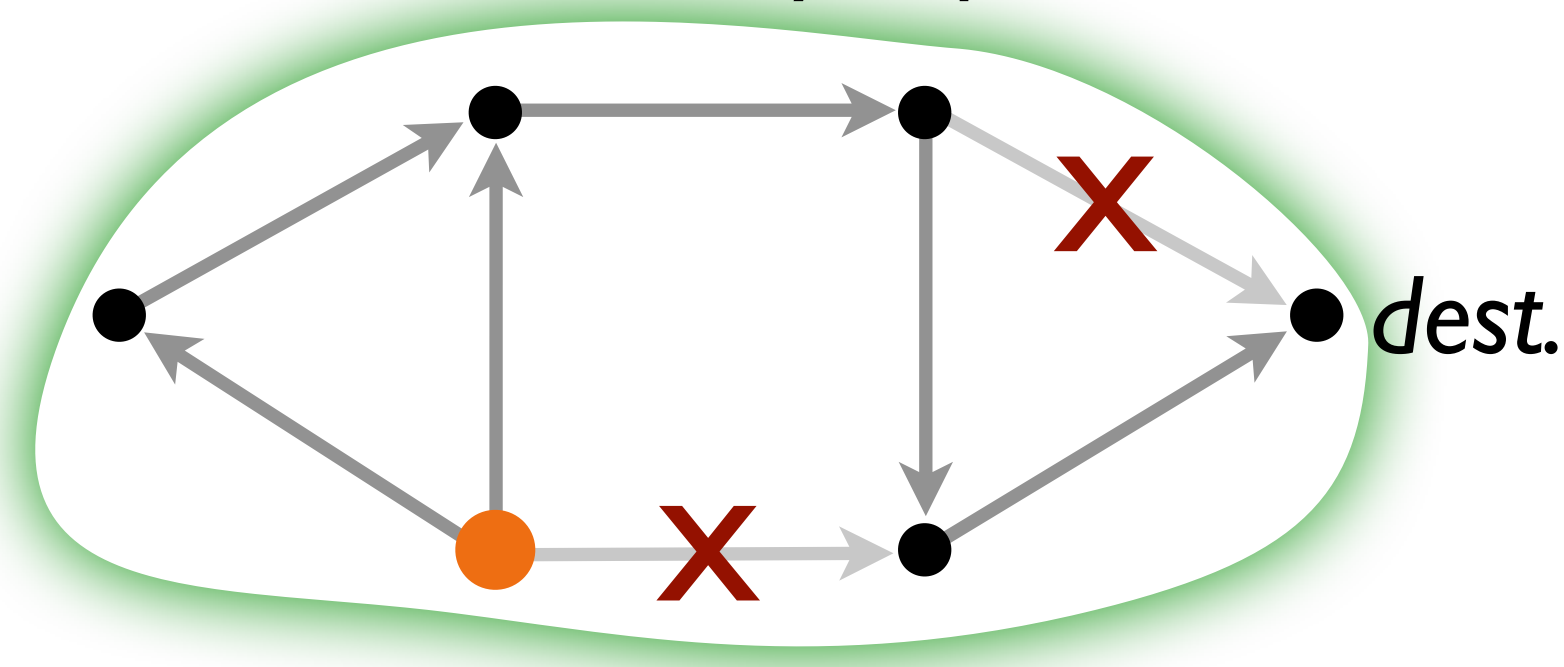
- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Guaranteed to converge

Define stable node: no more reversals

- If node *x* reverses adjacent to stable node *y*, then *x* also becomes stable
- Thus the stable set eventually expands to include all



Whew! Done!

Done!! ... please?

No: Protocol not yet suitable for the data plane

To reverse:

- Router must create new messages for each link
- Assumed these control messages arrive instantly and reliably
- Always have perfect information about distributed state!

Back where we started?

DDC: LR in the data plane

Data-Driven Connectivity
Liu, Panda, Singla, Godfrey, Schapira, Shenker
NSDI 2013

Key architectural point

- Make connectivity the job of the data plane
- Optimality (e.g. shortest paths) is still the job of the control plane

Problem

- LR requires sending control messages & distributed agreement on link direction – too slow for the data plane

DDC: LR in the data plane

Key algorithmic idea

- Allow stale info about link directions
- Each node can unilaterally reverse; notify neighbors later!
- Notification piggybacked on data packets using one-bit version number

Properties

- Strangely, this works...
- All events triggered by pkt arrival; no extra pkts created
- Simple bit manipulation operations

Stretch: DDC vs. MPLS

No guarantees on stretch, but empirically it's good

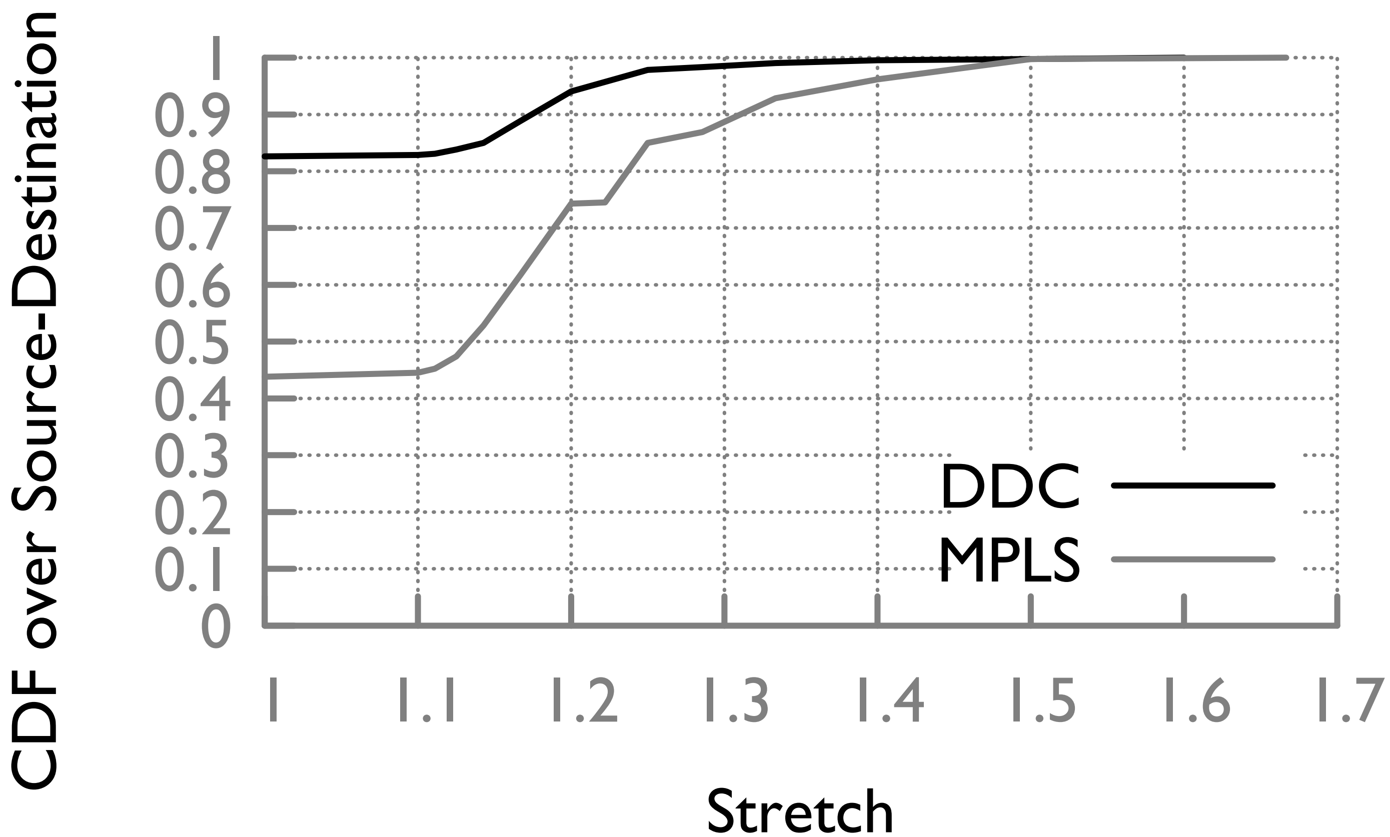


Figure 3: CDF of steady state stretch for MPLS FRR and DDC in AS2914.

Take-aways

The Internet is messy and opaque

- Empirically, unreliability is common
- End-to-end measurements provide a view "inside the black box"

Highly reliable routing is possible

- requires failure response in the data plane
- single-failure protection practical, with backup paths
- surprisingly, ideal connectivity is achievable

Announcements

Wednesday

- Selfish agents

Watch for survey this week

- Select special topics
- General feedback

After spring break

- Project midterm presentations
- Course staff is here to help – book time with one of us!