ARIADNE
Agnostic **R**econfiguration **I**n **A** **D**isconnected **N**etwork **E**nvironment

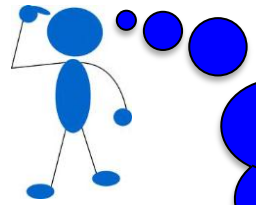
Konstantinos Aisopos (Princeton, MIT), Andrew DeOrio (Michigan),
Li-Shiuan Peh (MIT), Valeria Bertacco (Michigan)

What is “reconfiguration”?

Silicon technologies move into the nanometer regime

…transistors become unreliable

for architects: permanent faults

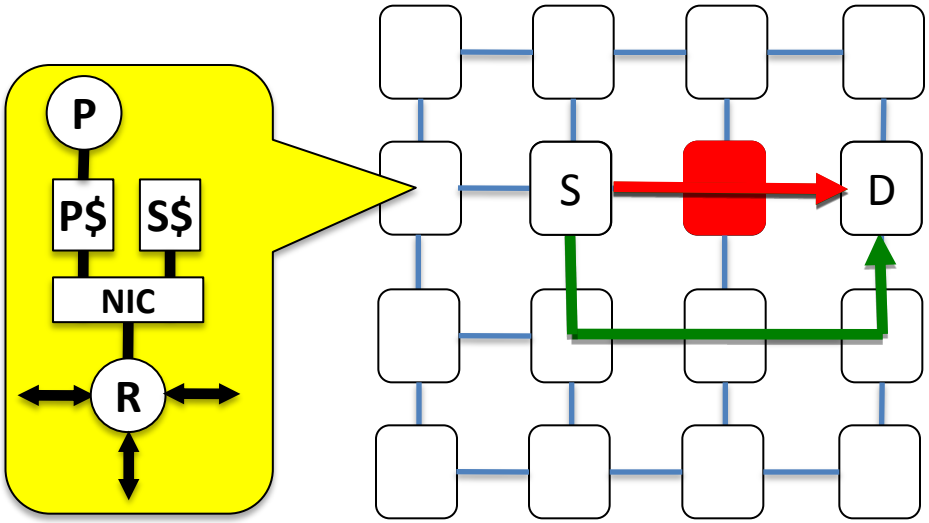


Shekhar Borkar
(Intel Fellow)

In future chips of 100 billion transistors, 10% of transistors will eventually fail over the lifetime of the chip

Our focus in this talk:

Network-on-Chip



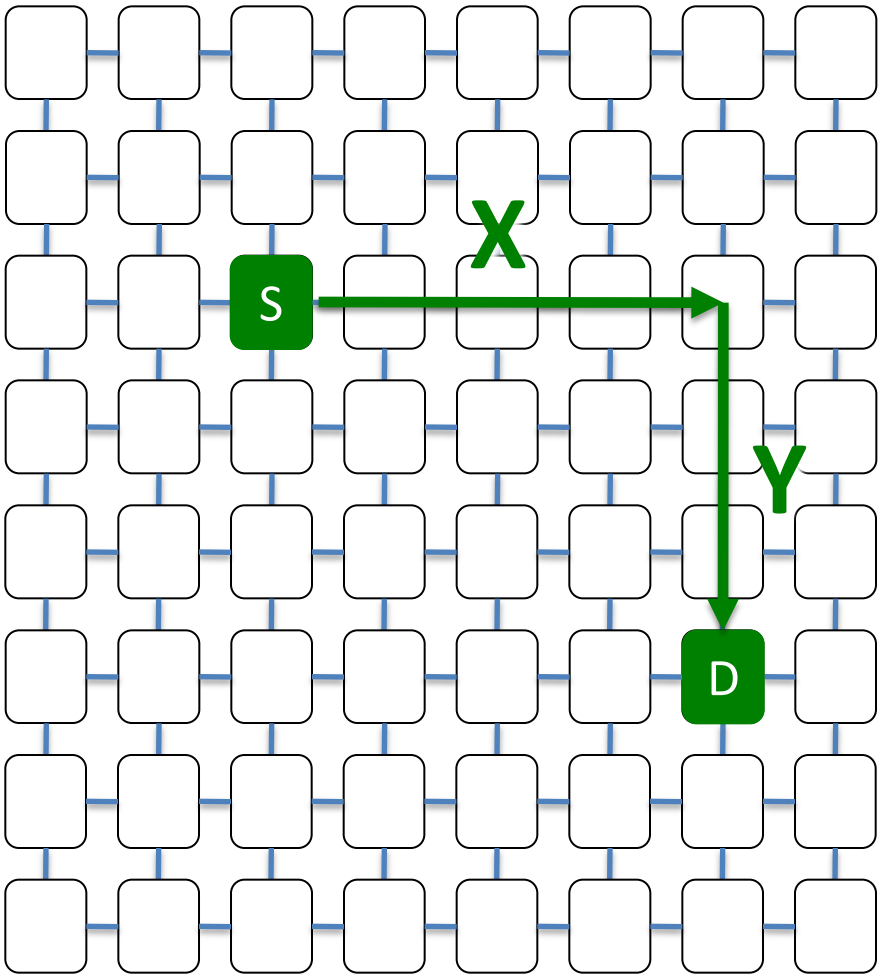
cannot resend

need to re-route around the fault

reconfiguration:
“the process of replacing the routing algorithm”

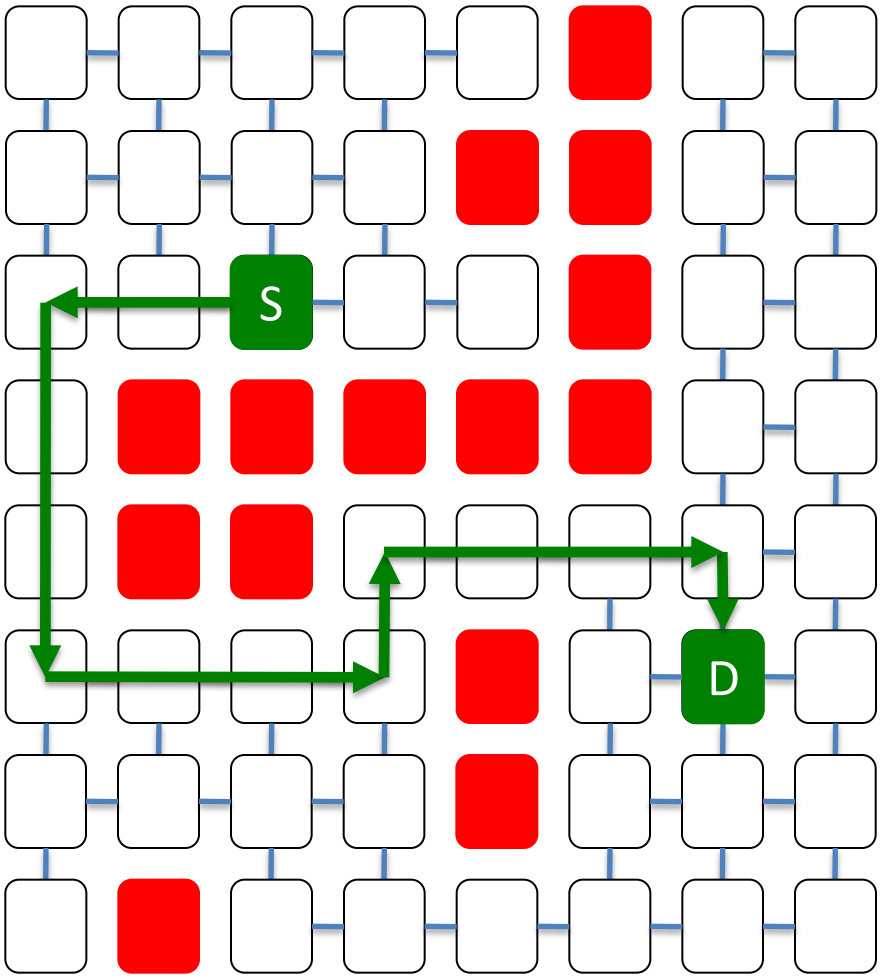
Why is reconfiguration challenging?

- XY routing



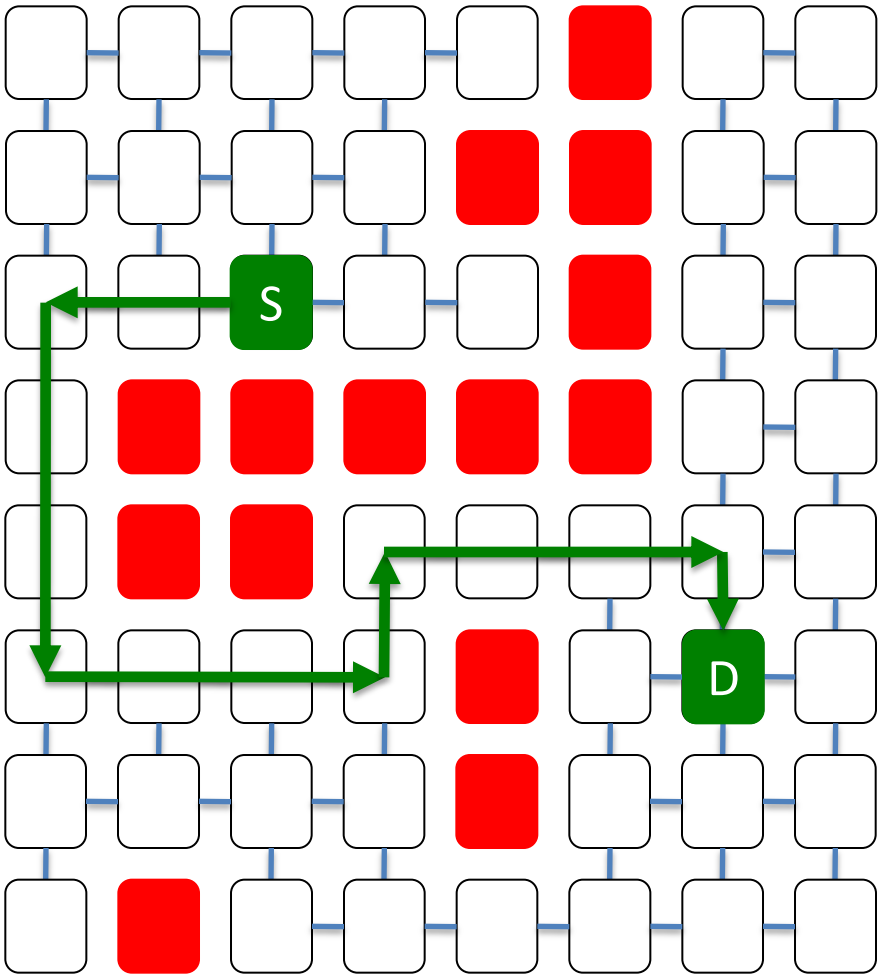
Why is reconfiguration challenging?

- ✗ XY routing
- ✓ Agnostic Reconfiguration algorithm
- ✓ In A Disconnected Network Environment



Why is reconfiguration challenging?

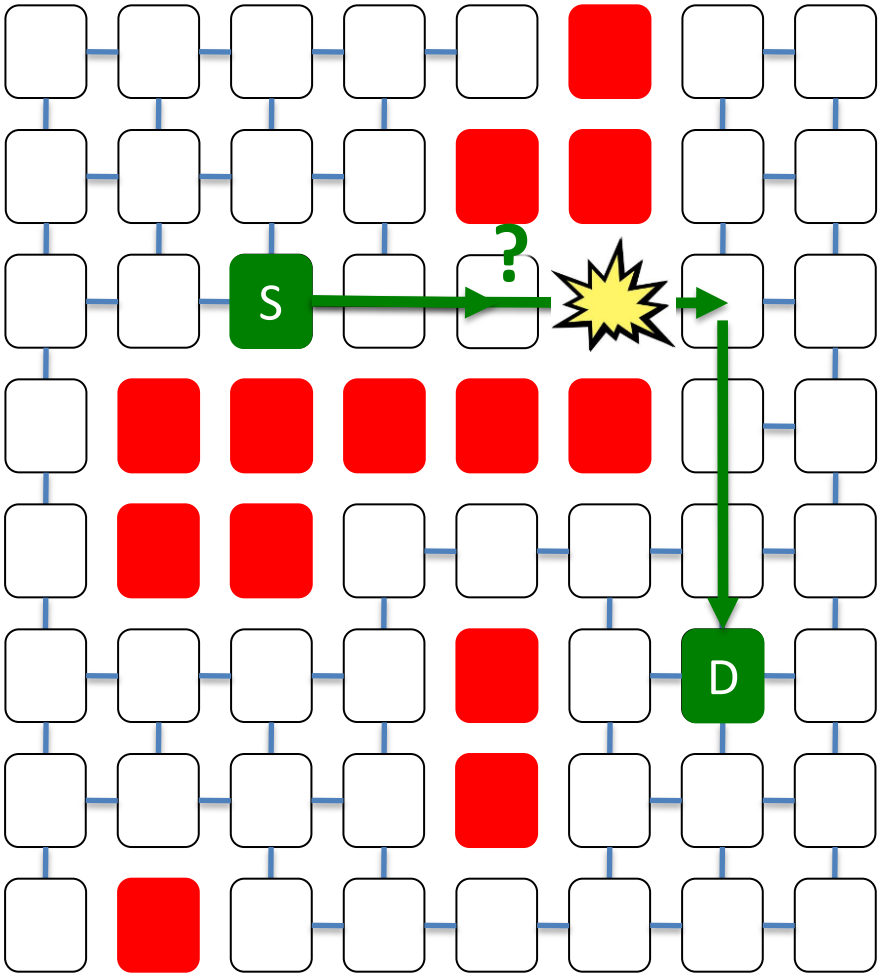
- ✗ XY routing
- ✓ **A**gnostic **R**econfiguration algorithm
- ✓ **I**n **A** **D**isconnected **N**etwork **E**nvironment



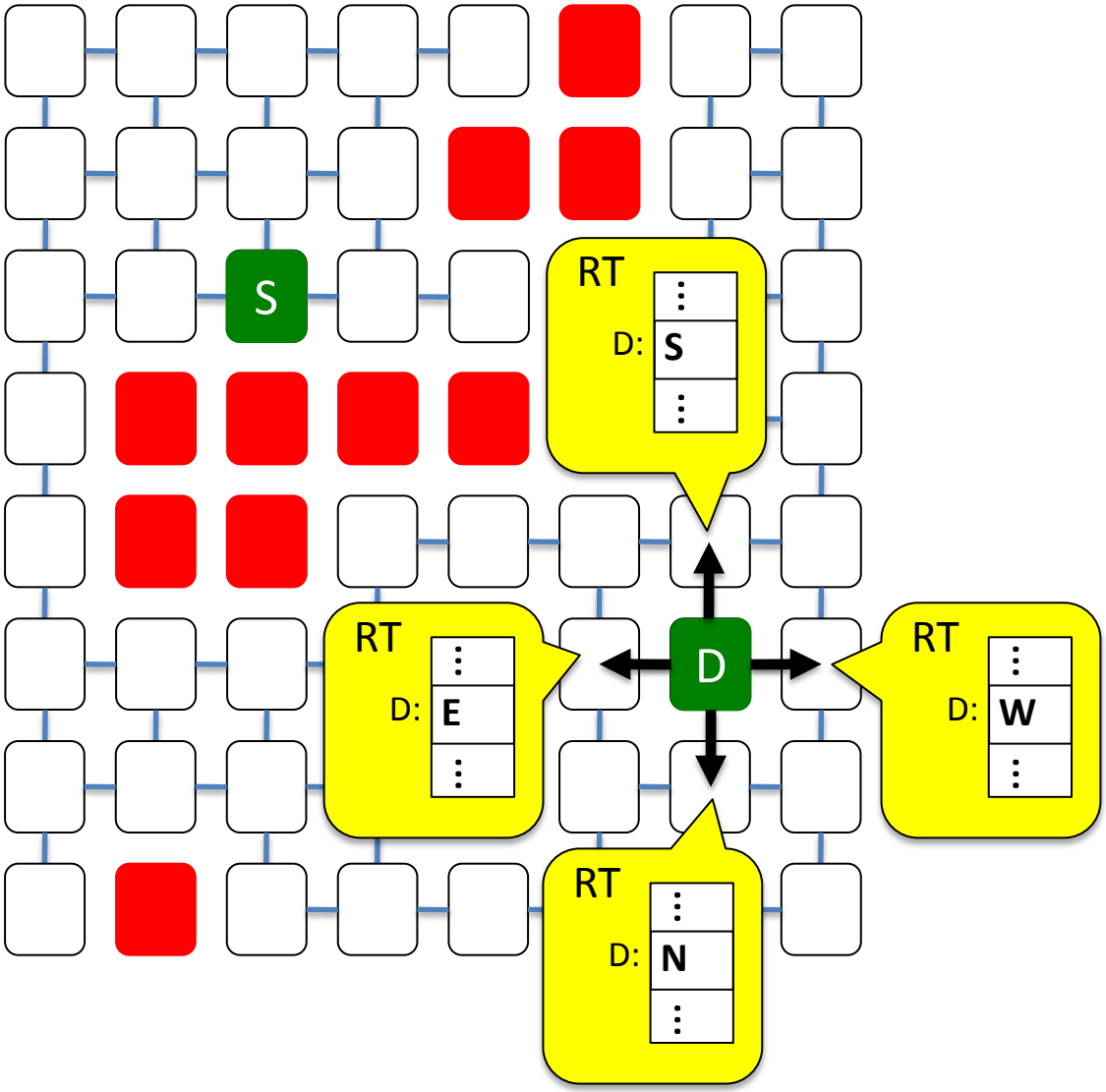
Outline

- Motivation
- Ariadne
  - Baseline
  - Deadlocks
  - Synchronization
- Evaluation
  - Overhead
  - Performance
  - Reliability
- Conclusions

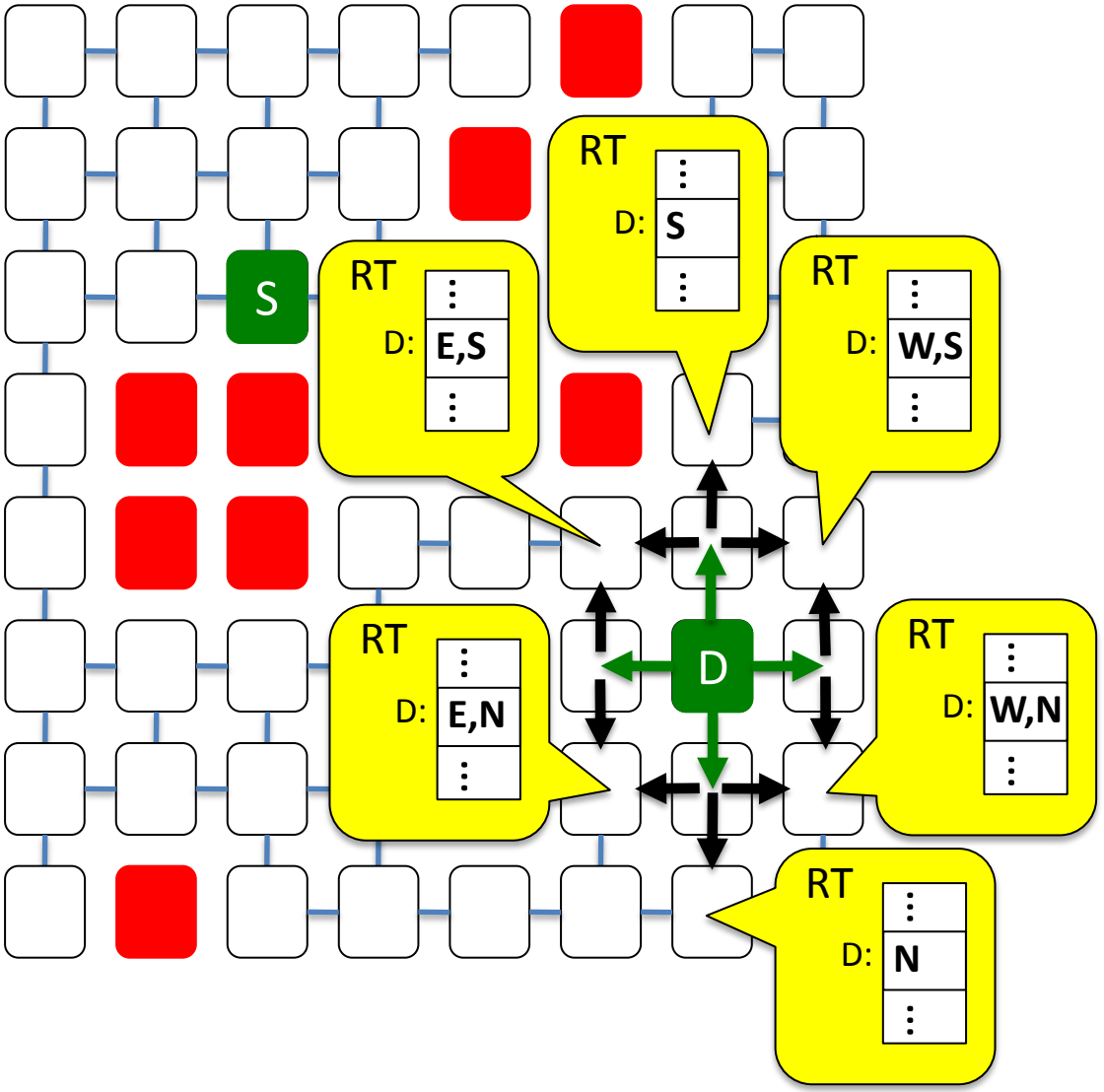
How will S find a path to D?



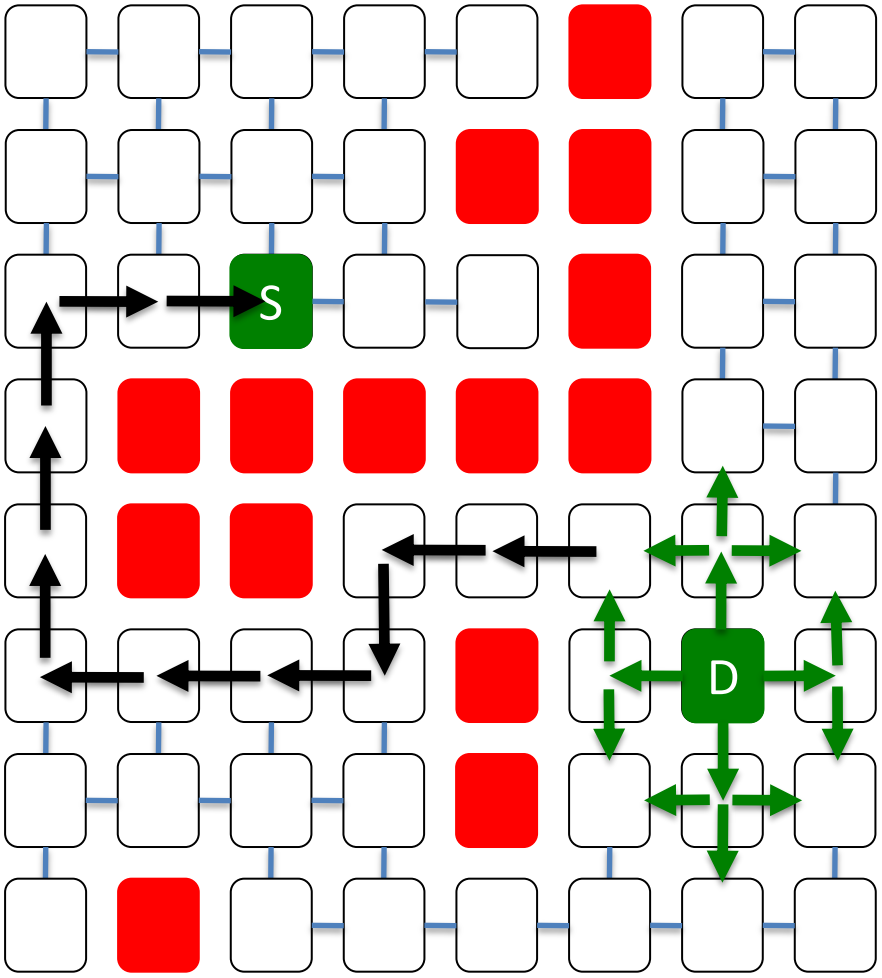
How will S find a path to D?



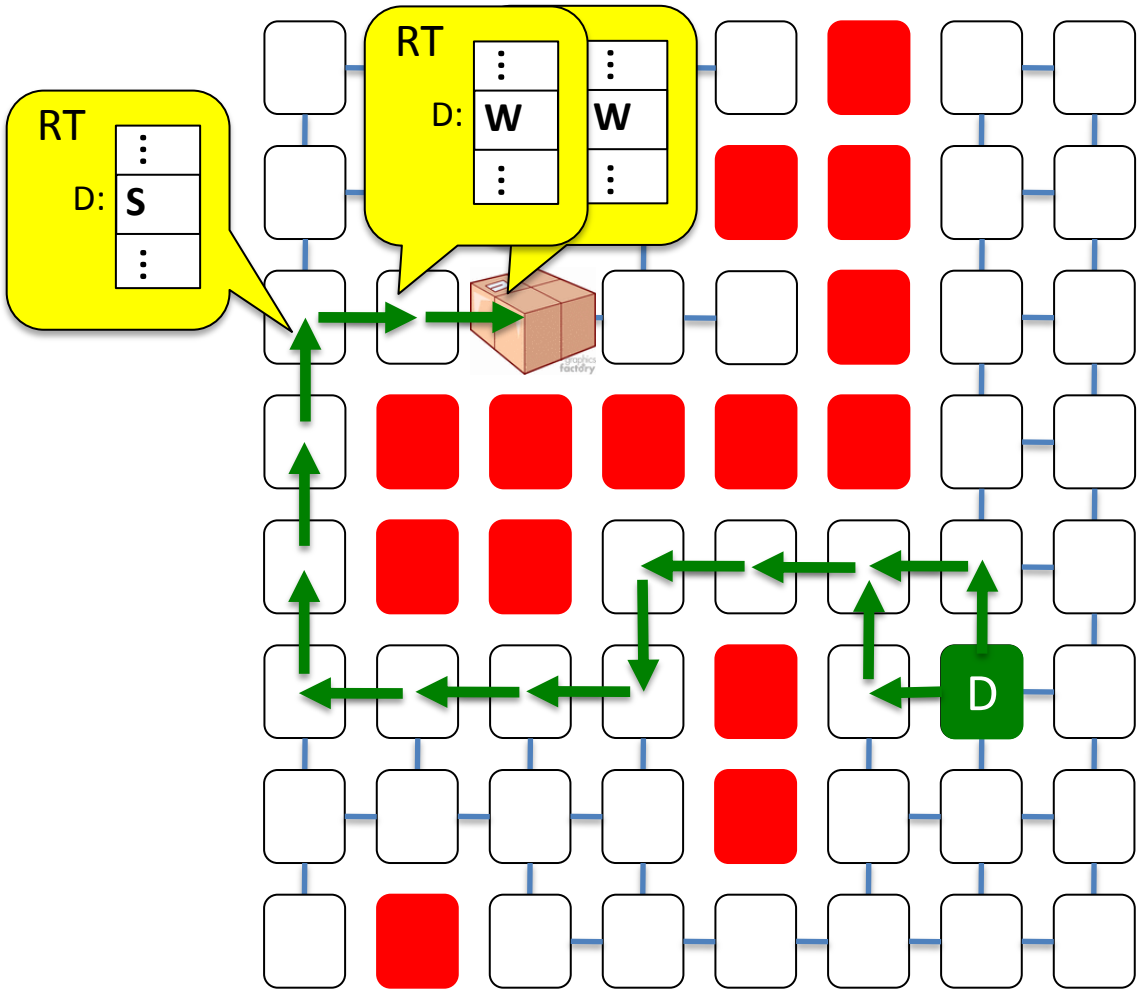
How will S find a path to D?



How will S find a path to D?



How will S find a path to D?

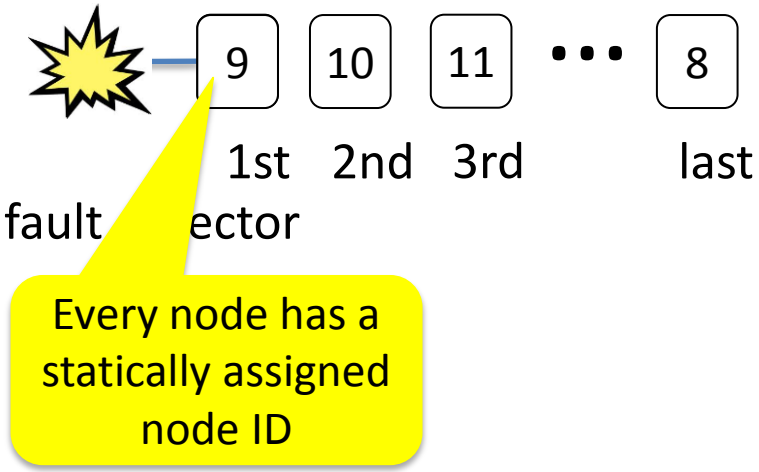


ARIADNE: baseline

- **Upon a fault that changes the topology…**
  - a node can let everyone know how it can be reached with a single broadcast
  - N nodes can let everyone know how they can be reached with N broadcasts

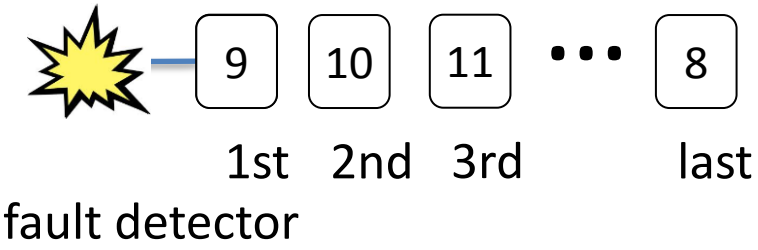
ARIADNE: baseline

- **Upon a fault that changes the topology…**
  - **Every node broadcasts "in-turn" to let others know how it can be reached**



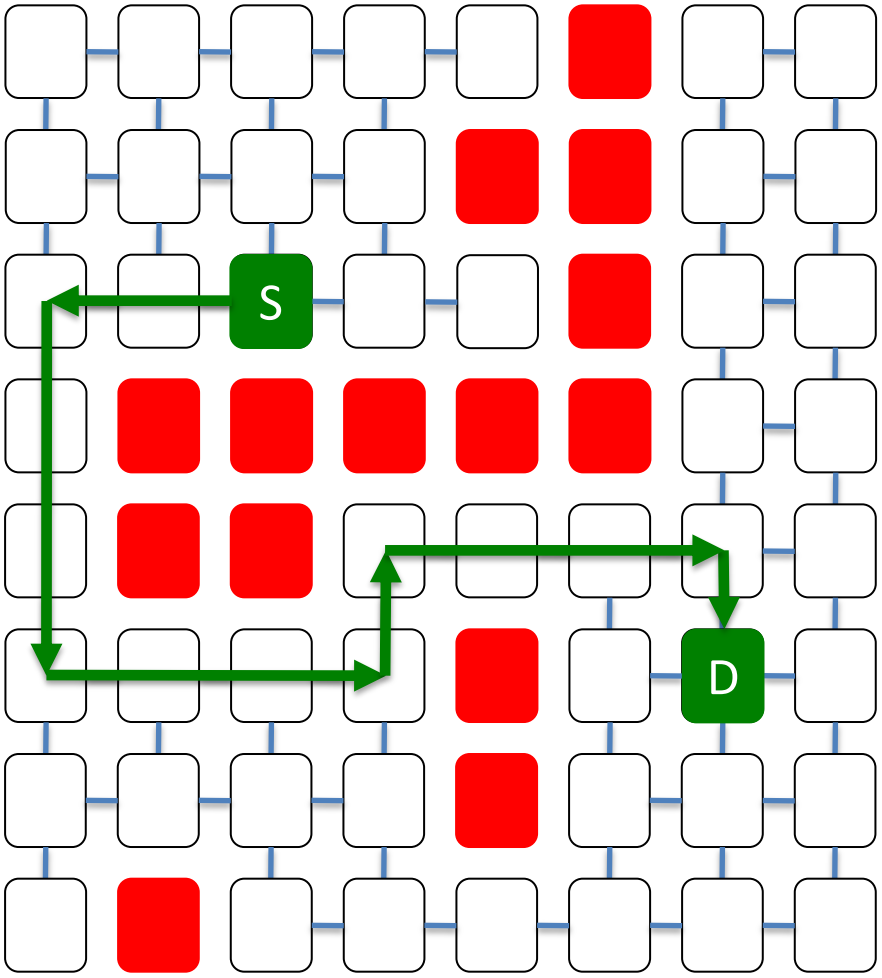
ARIADNE: baseline

- **Upon a fault that changes the topology…**
  - **Every node broadcasts "in-turn" to let others know how it can be reached**

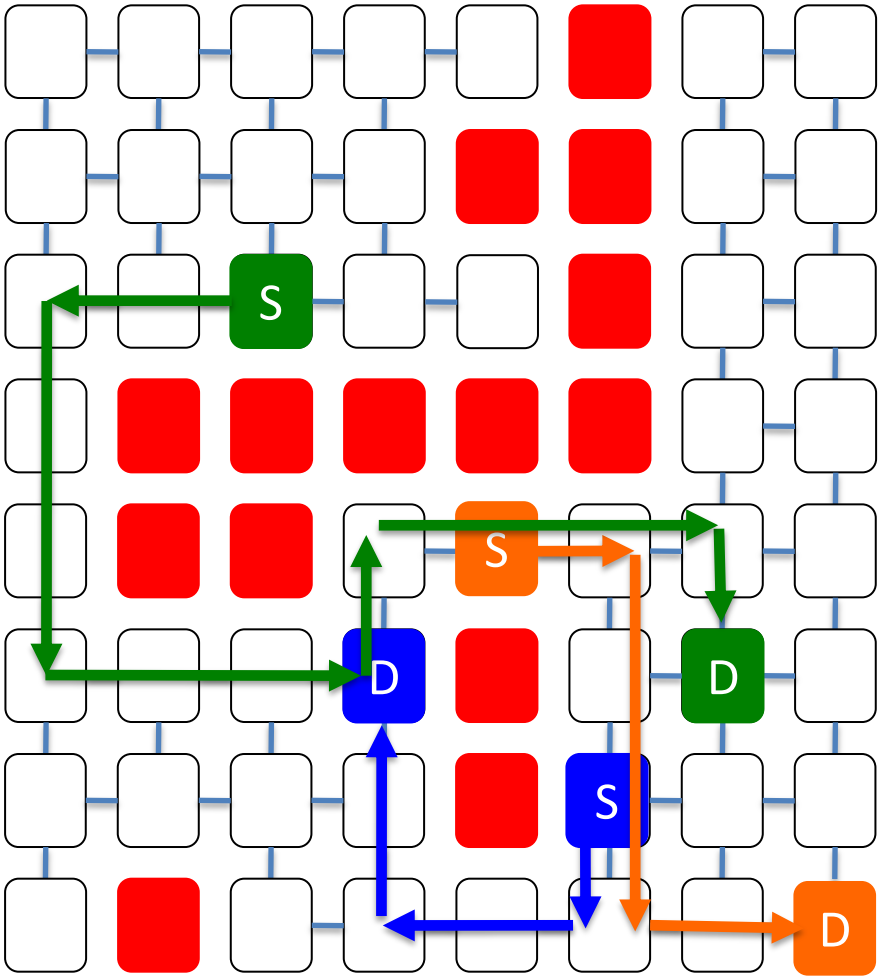


- **Issues:**
  - **deadlock avoidance**
  - **synchronization** (when to broadcast, multiple detectors)

ARIADNE: deadlocks



ARIADNE: deadlocks



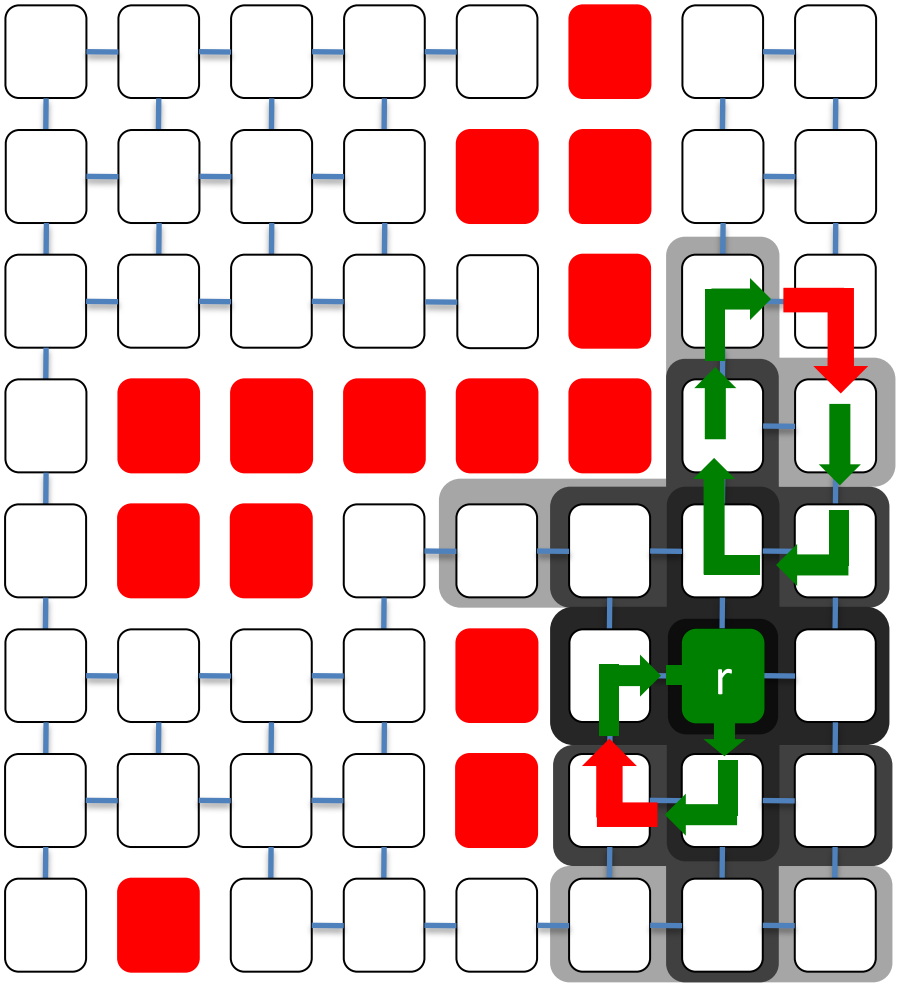
ARIADNE: deadlocks

up*/down*

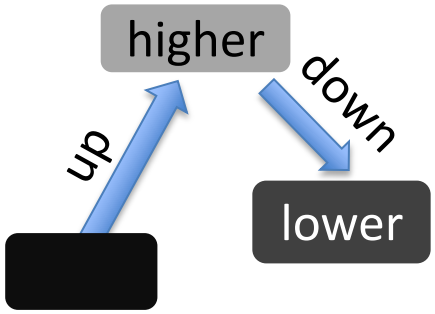
first bcast ONLY: nodes are assigned ranks

- 0 — bcaster "root"
- 1 — immediate neighbors
- 2 — 2-hop neighbors
- 3 — 3-hop neighbors

unique ordering: among nodes with same rank, arbitrarily select a higher one



disable routes where rank goes



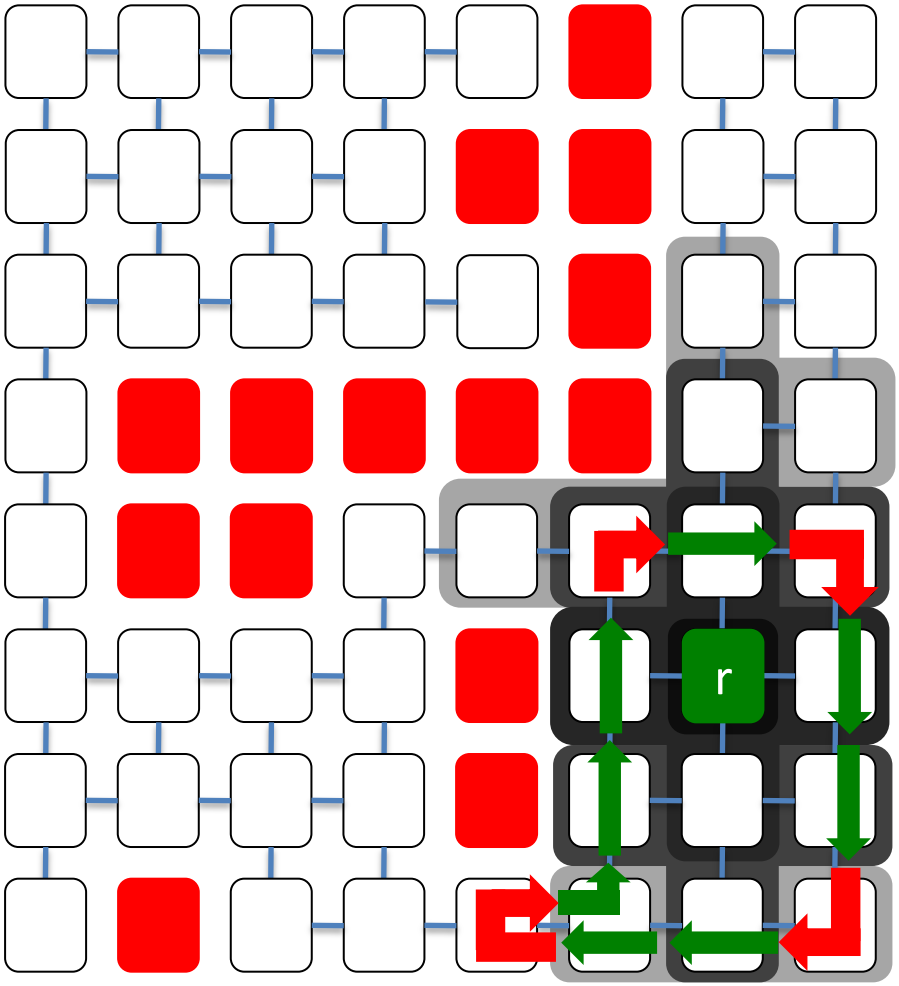
in every circle: 1 node will have higher rank than its neighbors, breaking the circular route

ARIADNE: deadlocks

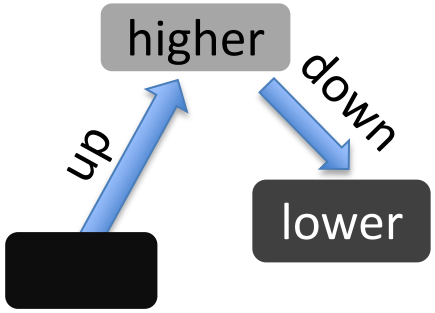
<u>up*/down*</u>
first bcast ONLY: nodes are assigned ranks

- 0 — bcaster "root"
- 1 — immediate neighbors
- 2 — 2-hop neighbors
- 3 — 3-hop neighbors

<u>unique ordering</u>: among nodes with same rank, arbitrarily select a higher one



disable routes where rank goes



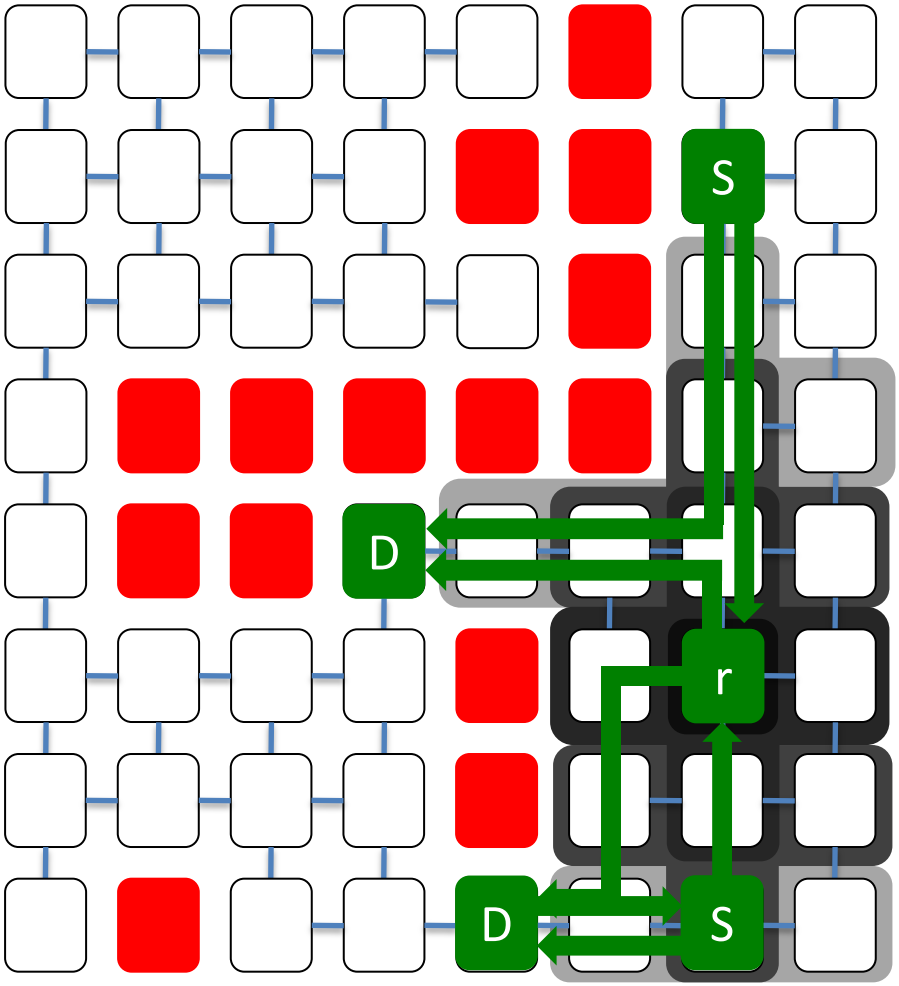
in every circle:
1 node will have higher rank than its neighbors, breaking the circular route

ARIADNE: deadlocks

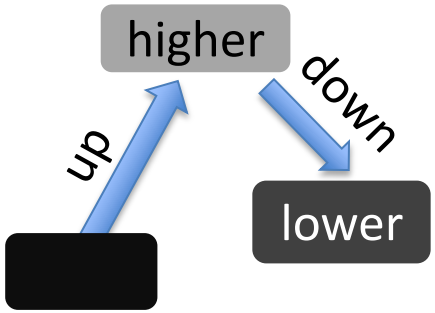
<u>up*/down*</u>
first bcast ONLY: nodes are assigned ranks

- 0 — bcaster "root"
- 1 — immediate neighbors
- 2 — 2-hop neighbors
- 3 — 3-hop neighbors

<u>unique ordering</u>: among nodes with same rank, arbitrarily select a higher one



disable routes where rank goes



in every circle: 1 node will have higher rank than its neighbors, breaking the circular route

<u>connectivity</u>: can reach any node via the root

ARIADNE: deadlocks

- **Upon a fault that changes the topology…**
  - **Every node broadcasts "in-turn" to let others know how it can be reached**

- **Issues:**
  - **deadlock avoidance** ✓
  - **synchronization**

ARIADNE: deadlocks

- **Upon a fault that changes the topology…**
  - **Every node broadcasts "in-turn" to let others know how it can be reached**
    RULE: (i) first broadcast ranks nodes
    (ii) remaining broadcasts spread only via enabled turns
- **Issues:**
  - **synchronization**

ARIADNE: synchronization

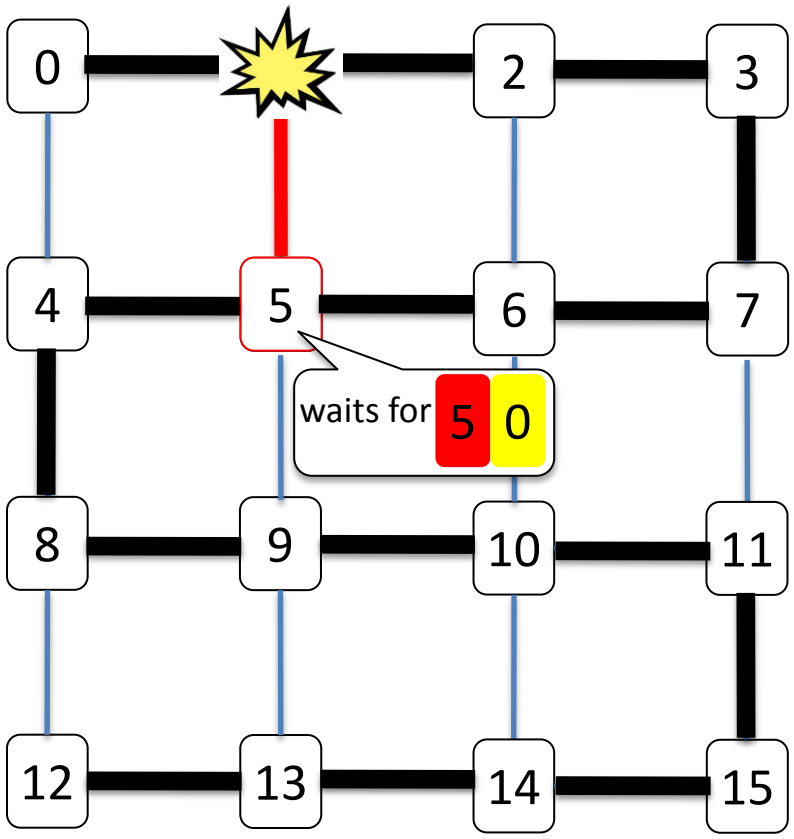
- How do I know completion of previous broadcast? can broadcasts overlap?
- How does the recipient of a flag know the broadcasting node?

ARIADNE: synchronization

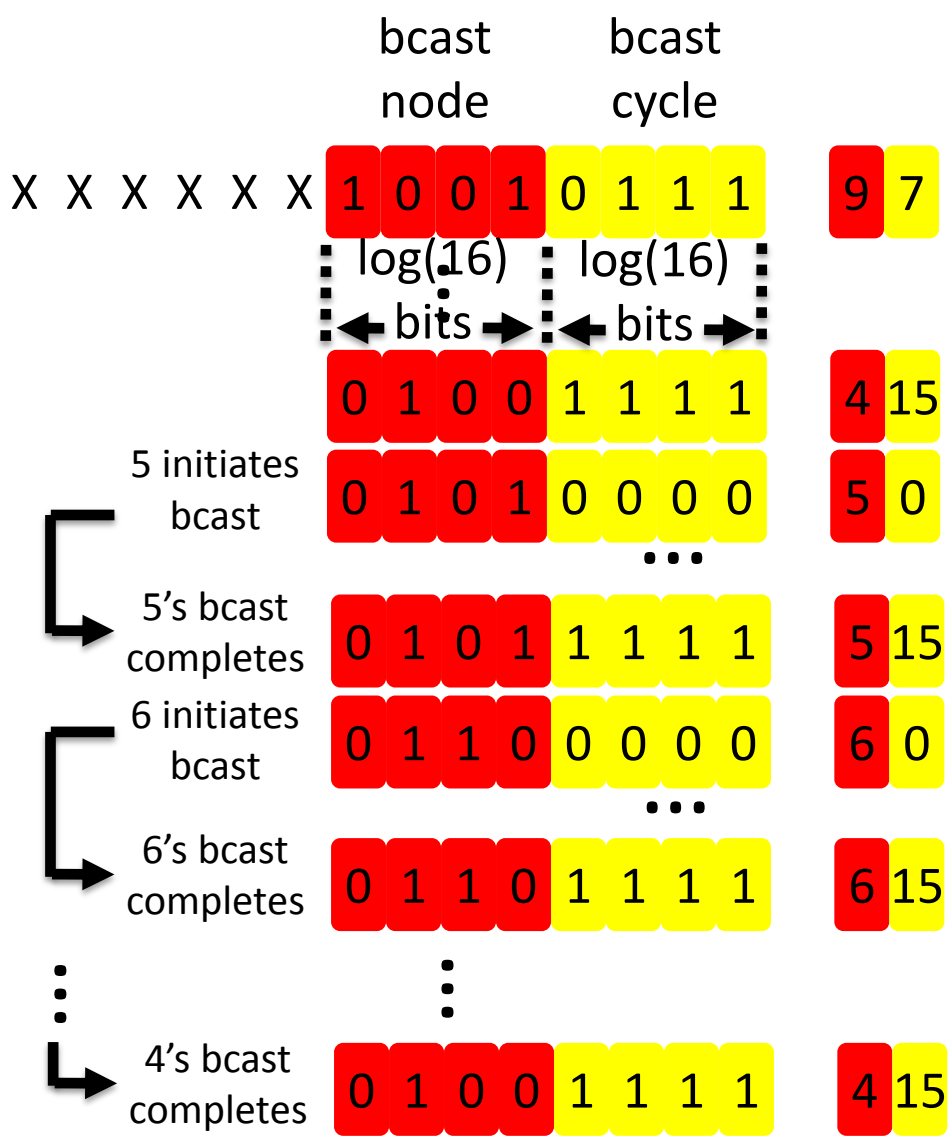
Solution : Atomic Broadcasts

- Nodes utilize the cycle count as a global reference point
- Each node is assigned a unique broadcast slot from the “global” cycle counter

ARIADNE: synchronization

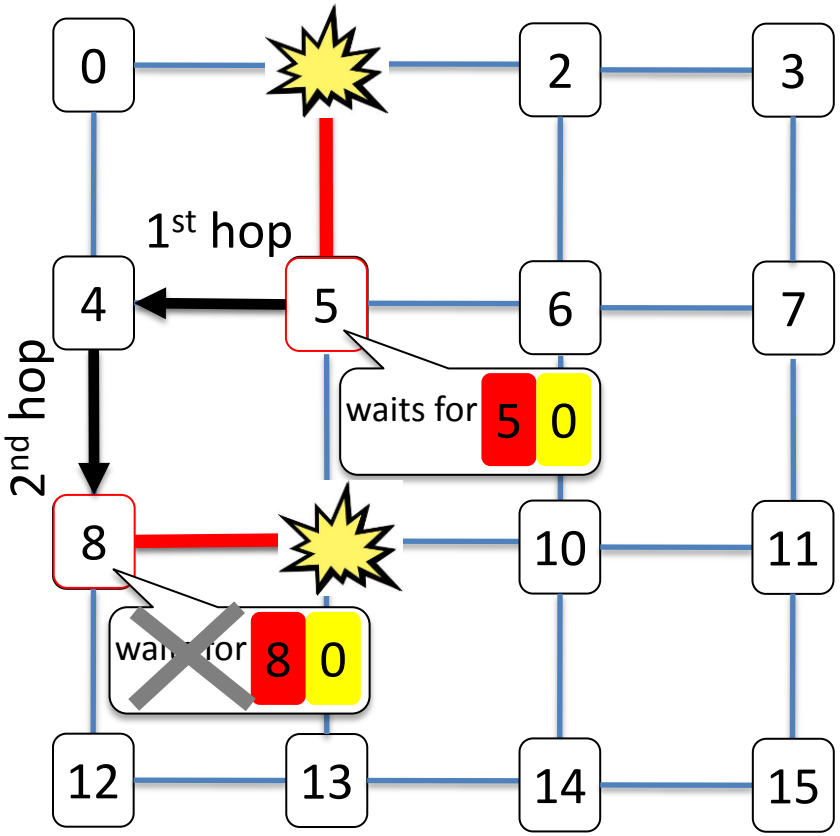


longest (in hops) broadcast

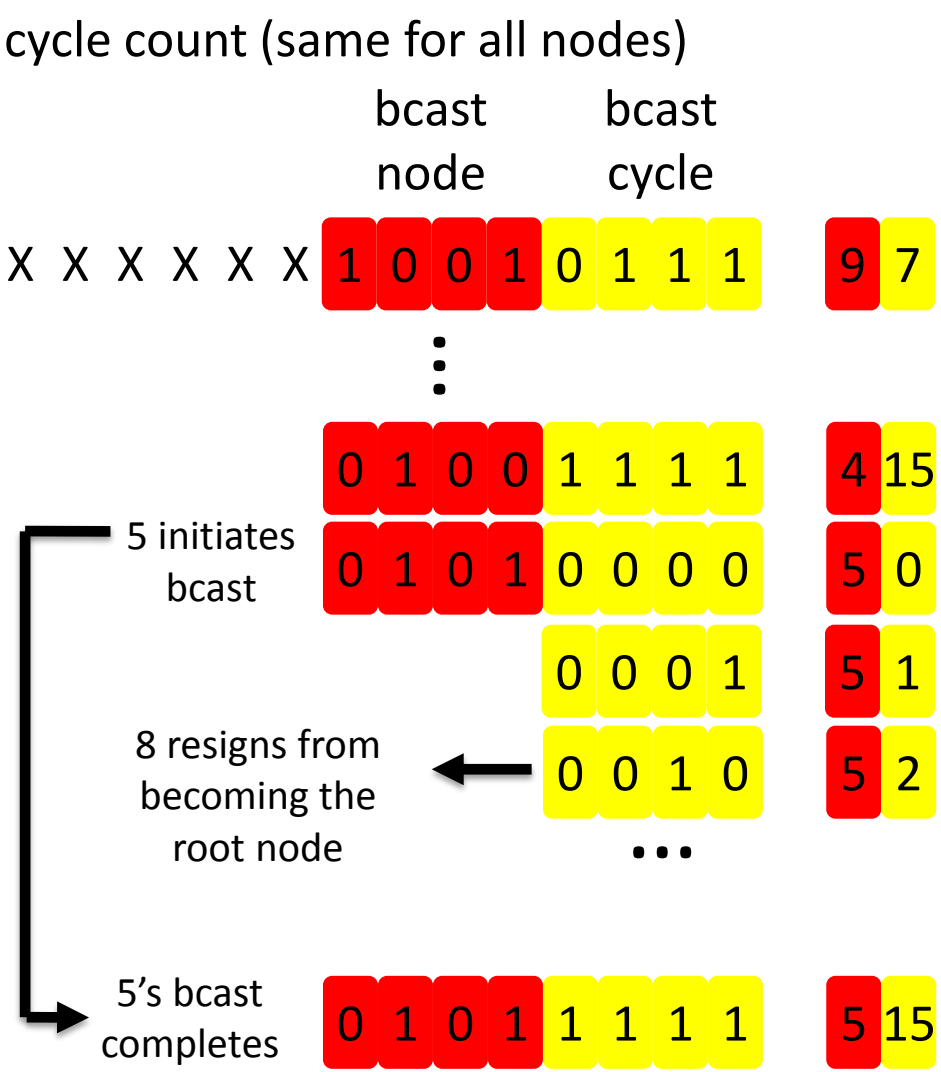


reconfiguration completes in
$(16)^2$ =(number of nodes)^2 cycles

ARIADNE: synchronization



(!) we need to reconfigure once even for multiple faults



Outline

- Motivation
- Ariadne
  - Baseline
  - Deadlocks
  - Synchronization
- Evaluation
  - Overhead
  - Performance
  - Reliability
- Conclusions

Evaluation: Overhead

- On-chip routing algorithms for irregular topologies

	ARIADNE	Immunet (V. Puente, ISCA'04) reserves an escape VC for deadlock freedom (routes deterministically in a ring)	Vicis routing algo (D. Fick, DATE'09) exceptions to turn model to apply it to an arbitrary topology
overhead	✓ 2.0%	✗ 6.0%	✓ 1.5%
performance			
reliability			

synthesized a baseline 5-stage pipelined router (5 ports, 2 VCs, 5-flit buffer/VC)
with Synopsys Design Compiler (IBM 130nm target library):
router area (mm^2): baseline=2.708, Ariadne=2.761, Vicis=2.748, Immunet=2.870

Evaluation: Performance

- Experimental Setup:
  Garnet + GEMS

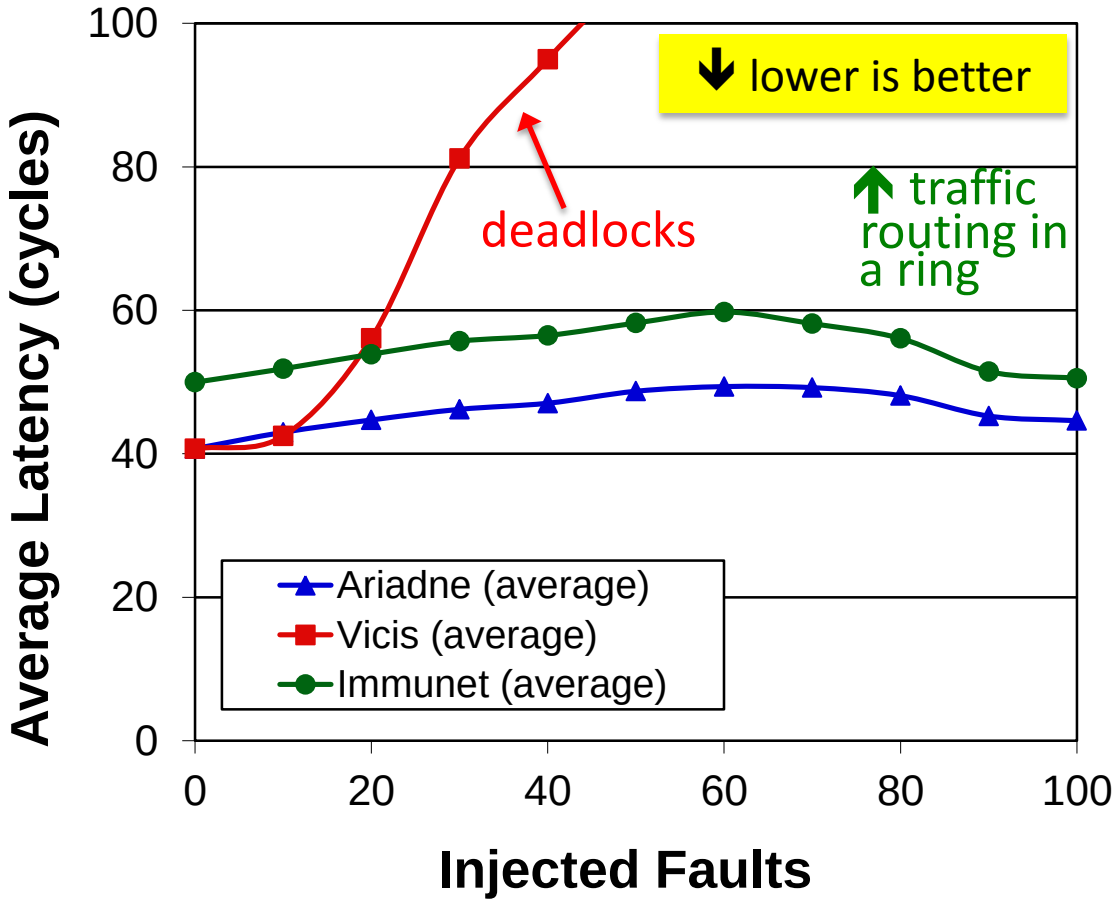
System Configuration (GEMS)

processors	In-order SPARC cores
coherence	MOESI protocol
L1 caching	private unified 32KB/node
	ways: 2 latency: 3 cycles
L2 caching	shared distributed 1MB/node
	ways: 16 latency: 15 cycles

Network Architecture (GARNET)

network topology	8x8 2D mesh
memory controllers	4 at chip corners
channel width	64 bits
router architecture	5-stage pipeline
router ports, VCs	5, 2 (private)
router buffers/port	5-flit for each VC

Average over 100 topologies
10 PARSEC benchmarks



Evaluation: Performance + Reliability

- On-chip routing algorithms for irregular topologies

	ARIADNE	Immunet (V. Puente, ISCA'04) reserves an escape VC for deadlock freedom (routes deterministically in a ring)	Vicis routing algo (D. Fick, DATE'09) exceptions to turn model to apply it to an arbitrary topology
overhead	✓ 2.0%	✗ 6.0%	✓ 1.5%
performance	✓	✗	✗
reliability	✓	✓	✗

Outline

- Motivation
- Ariadne
  - Baseline
  - Deadlocks
  - Synchronization
- Evaluation
  - Overhead
  - Performance
  - Reliability
- Conclusions

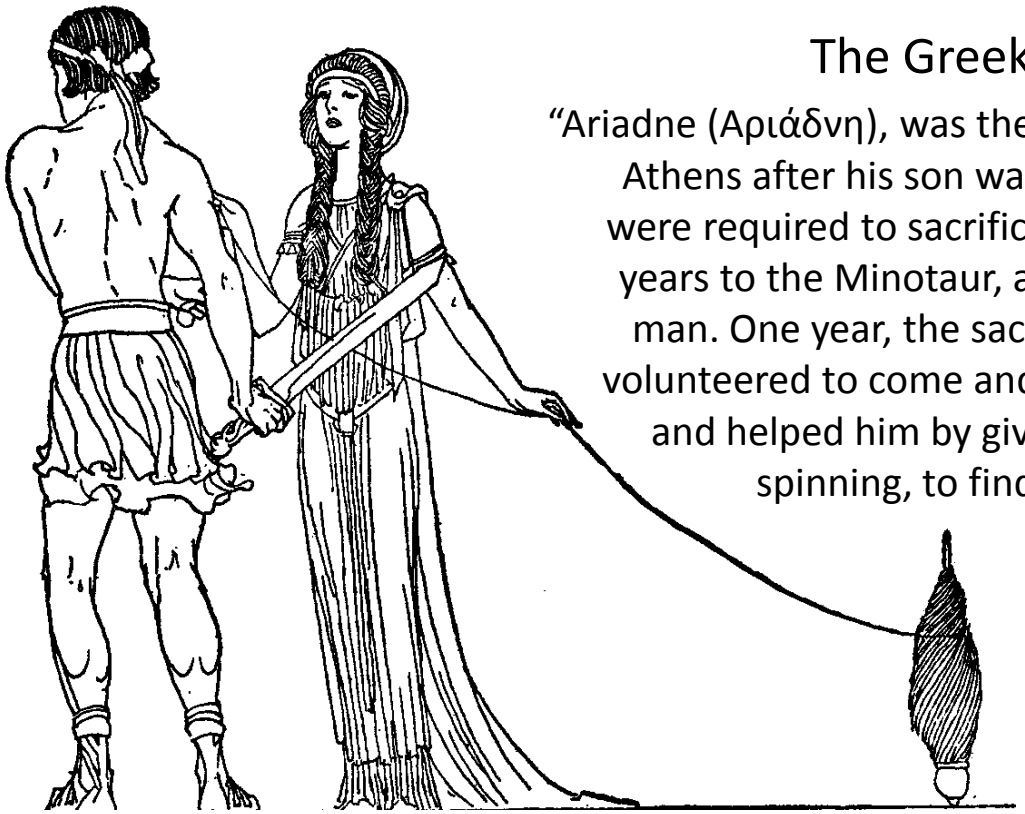
Conclusions

We have presented Ariadne.

- a reconfiguration algorithm that provides deadlock-free routing paths in irregular network topologies that result from faulty links
- is implemented in a fully distributed mode, resulting in simple hardware and low complexity
- enables a trade-off between performance and reliable functionality on unreliable silicon

Thank You!

Questions?



The Greek legend of Princess Ariadne [source: wikipedia]

"Ariadne (Αριάδνη), was the daughter of King Minos of Crete. Minos attacked Athens after his son was killed there. The Athenians asked for terms, and were required to sacrifice seven young men and seven maidens every nine years to the Minotaur, a monster with the head of a bull on the body of a man. One year, the sacrificial party included Theseus, a young man who volunteered to come and kill the Minotaur. Ariadne fell in love at first sight, and helped him by giving him a ball of red fleece thread that she was spinning, to find his way out of the Minotaur's labyrinth."

…similarly to Princess Ariadne, our Ariadne algorithm helps packets find their way in the labyrinth-like topology of a faulty network.