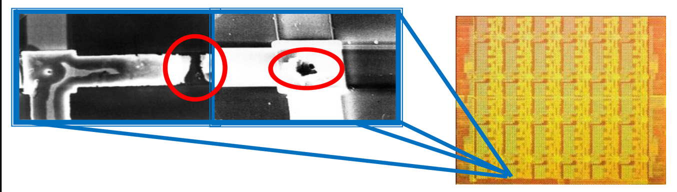
Unified Diagnosis and Reconfiguration for Frugal Bypass of NoC Faults

Ritesh Parikh and Valeria Bertacco

MICRO-46, December 7-11, Davis, CA

Permanent Faults in NoCs

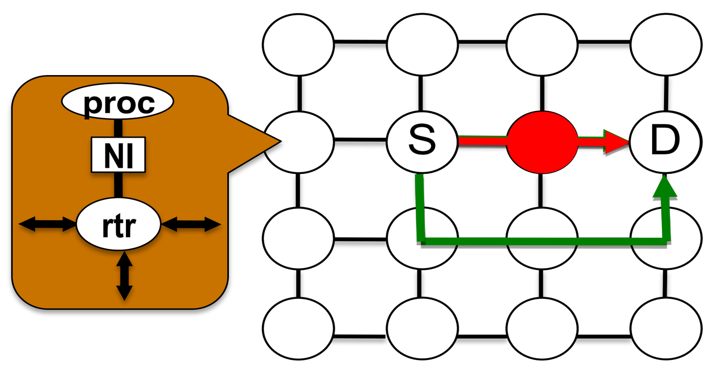


— Permanent wear-out
— Device aging

NoCs: significant silicon footprint and heavy activity

Detection → Diagnosis → Reconfiguration → Recovery

- Detection: Detect if fault has occurred
- Diagnosis: Diagnose where fault has occurred
- Reconfiguration: Reconfigure network to account for fault
- Recovery: Recover and resume normal operation

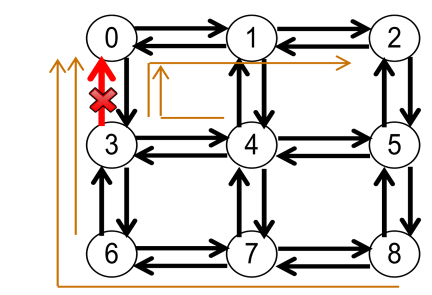


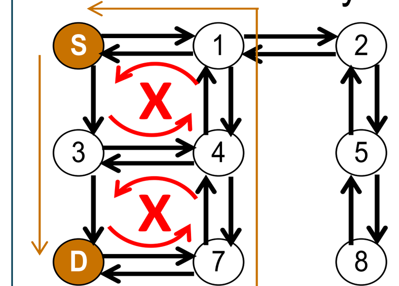
✗ cannot send on faulty path
✓ need to re-route around fault

uDIREC solves two problems:
- **Fault diagnosis** at a fine resolution
- **Reconfiguration** to find new routes

Unified Diagnosis and Reconfiguration

	Diagnosis	Reconfiguration
Past Work	—Entire regions/routers —One or more bidirectional links	— Based on routing on irregular networks — Constrained by number and location of faults
uDIREC	—End-to-end scheme in SW —Based on analyzing faulty routes —Passive and fine-grained	—Based on a novel routing algorithm —Tightly integrated with the diagnosis scheme —Unconstrained by number and location of faults

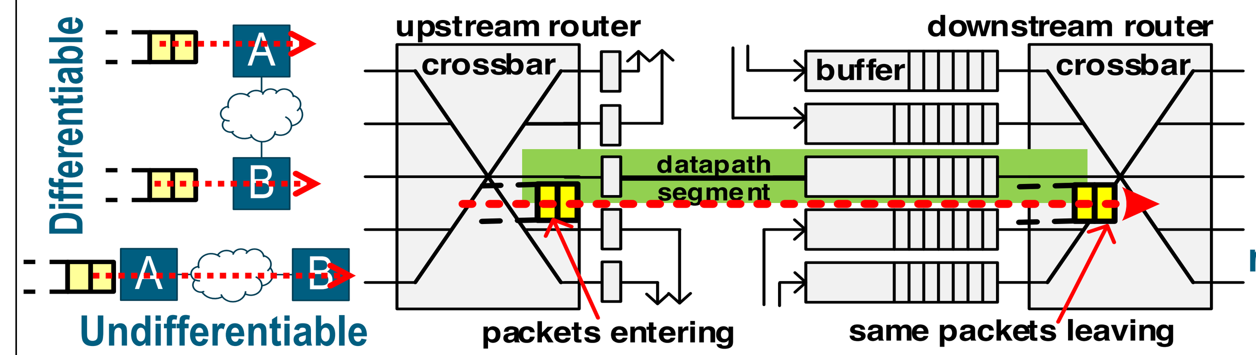




Faulty irregular network with deadlock-free routes

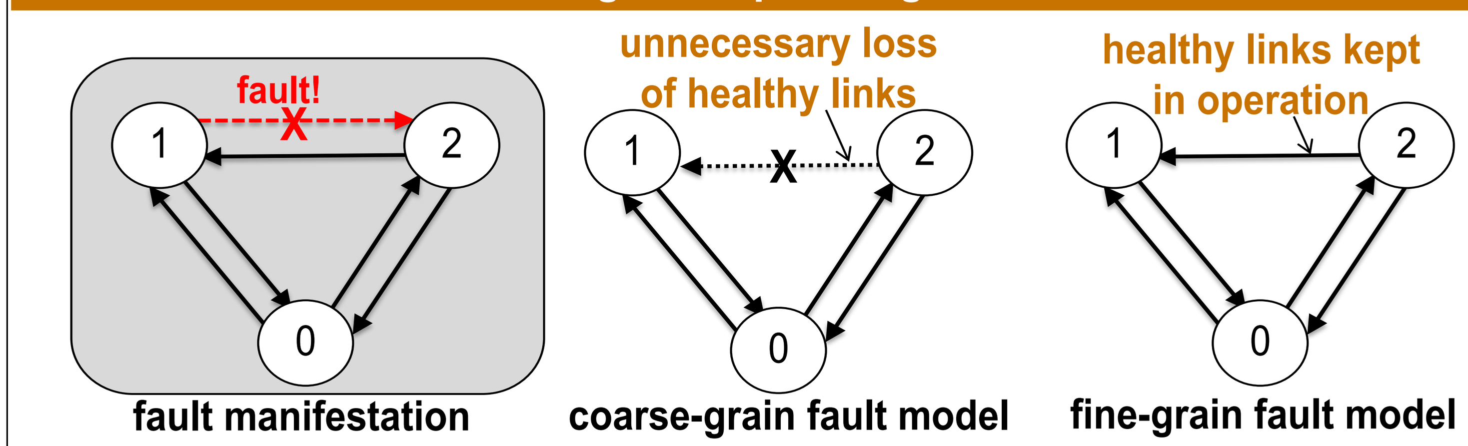
— **Dedicated testing is not required → no overhead in absence of errors**
— **Unified implementation in software → low area overhead**

Fine-Resolution Fault Diagnosis



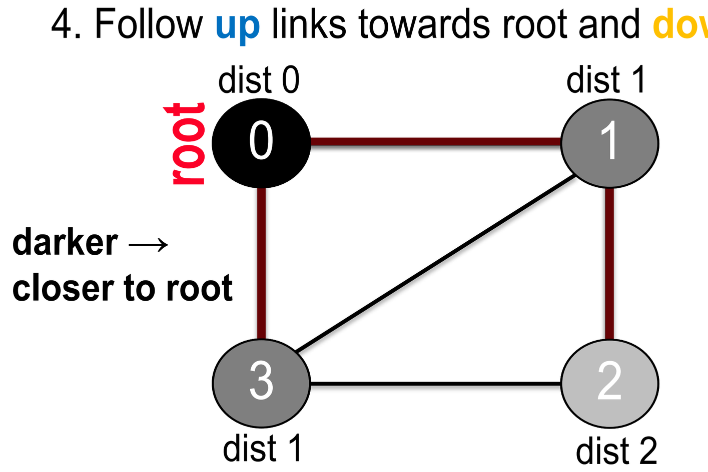
Finest granularity of diagnosis and reconfiguration

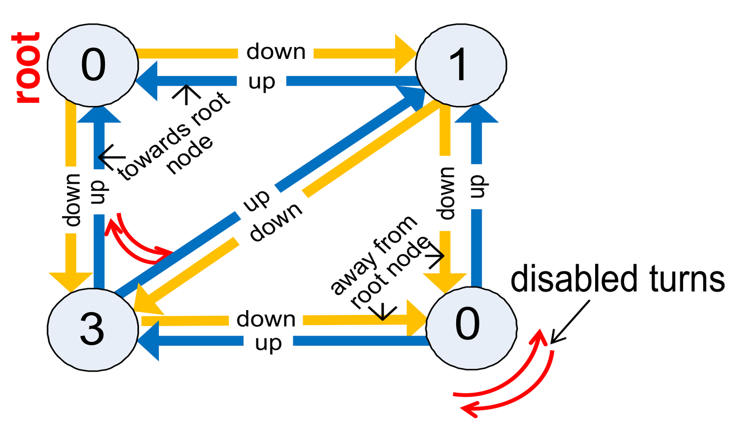
96% faults localized to a single datapath segment, or a unidirectional link



Routing in Irregular Networks

— Routing algorithm should disable paths that lead to deadlock
— Up*/down* routing disables **turns** to avoid deadlock
1. Construct **spanning tree** rooted at a node (assumes bidirectional links)
2. Mark links towards root: up (down otherwise)
3. Disable all down→up turns
4. Follow up links towards root and down links to destination



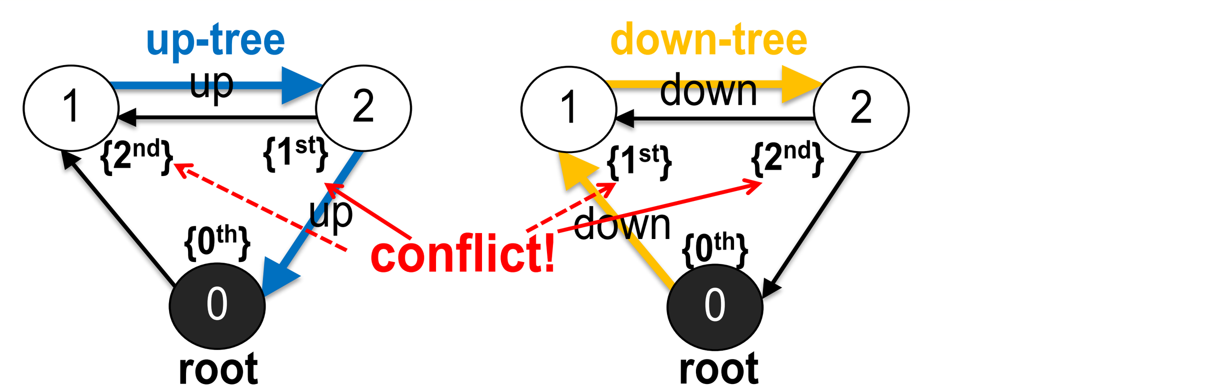


Up*/down* does not work with unidirectional links

Routing with Unidirectional Links

— Separate spanning trees for up (up-tree) and down (down-tree) links
- Up-tree: unidirectional links towards root
- Down-tree: unidirectional links away from root
- **Consistent ordering/labeling**: no link marked both up and down

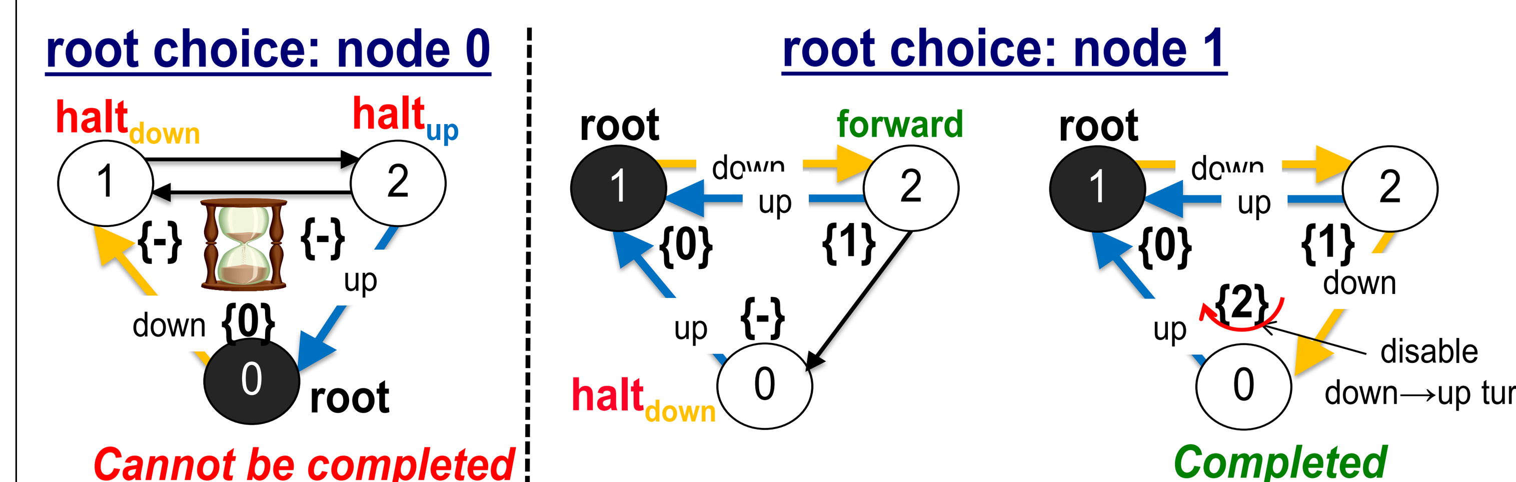
— As links are marked according to up*/down* principle (and no conflicts)
- uDIREC routing is **deadlock free** (disable down→up turns)
- Network is **connected** if both trees are complete



Growing Trees Concurrently

up-tree and down-tree can be constructed:
- **independently**: may lead to inconsistent marking
- **concurrently**: consistent labeling guaranteed by construction

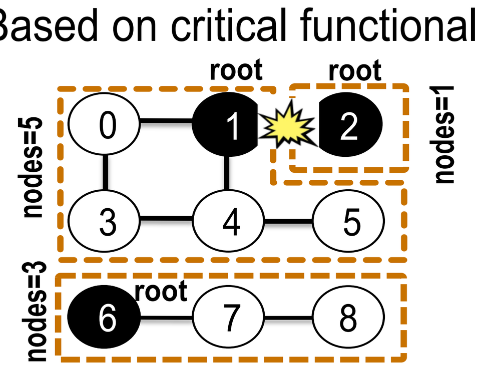
expand trees beyond a node only if reachable by both up-tree and down-tree



choice of root node affects connectivity

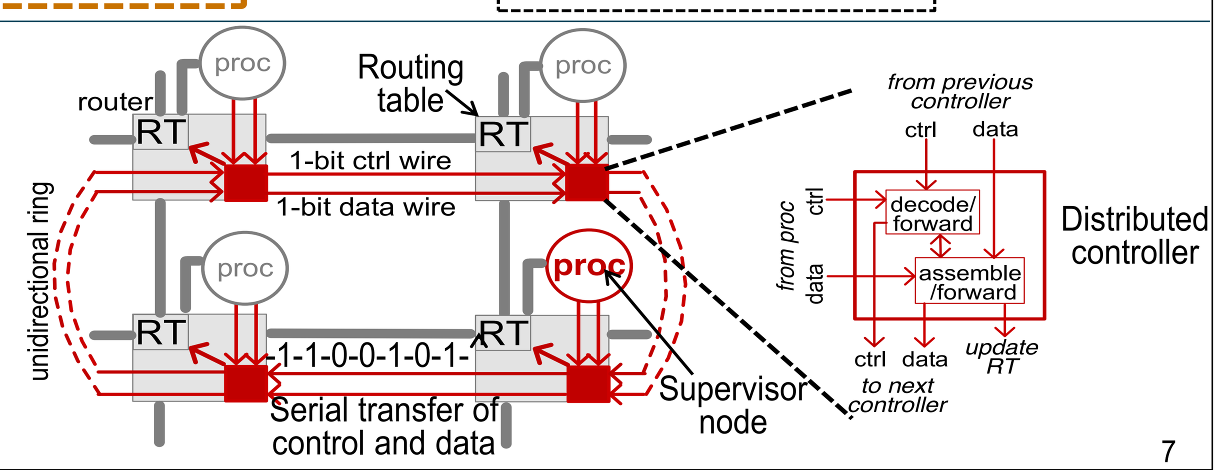
Reconfiguration Algorithm

— Exhaustive search of the optimal root node
— Based on critical functionality or number of connected nodes

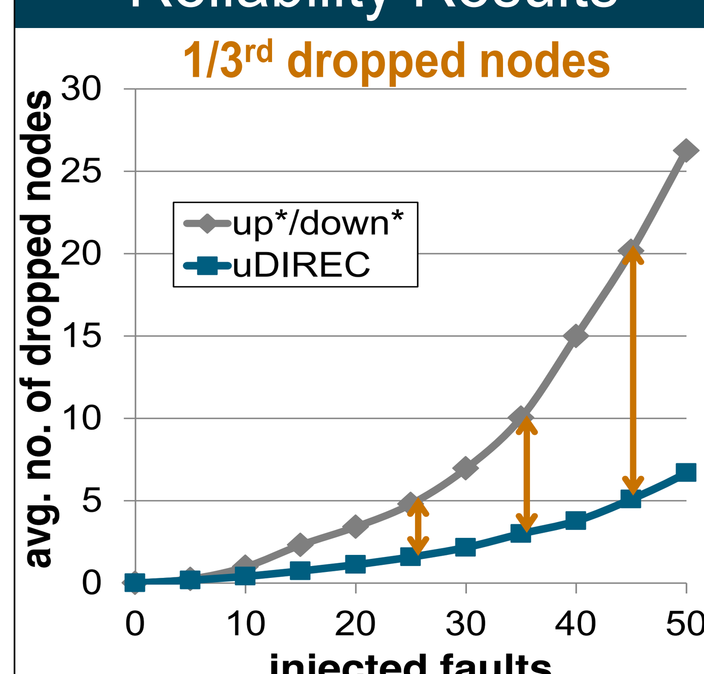


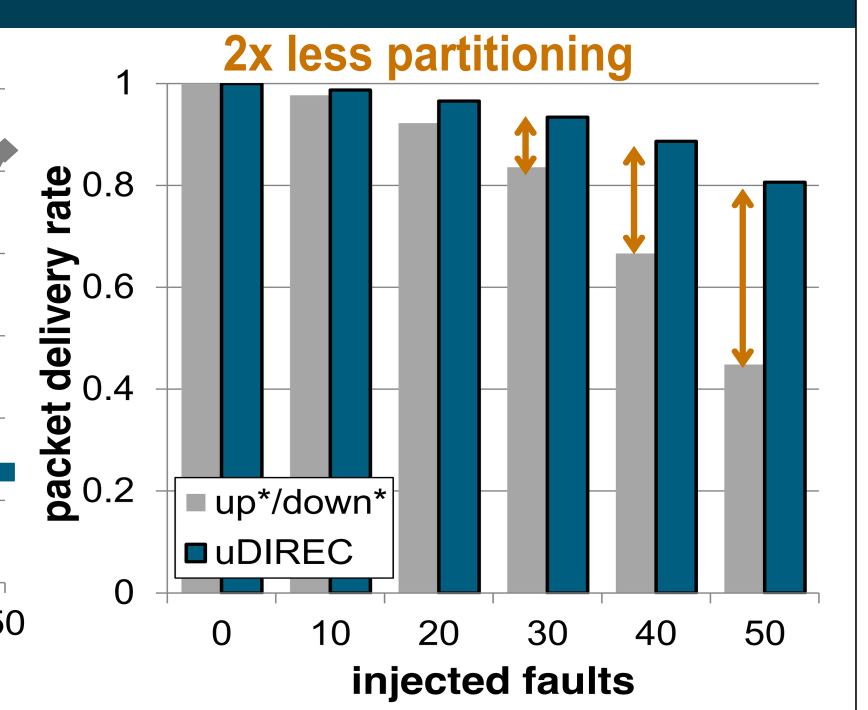
root trial #1: node 1 is root	local winner!
root trial #2: node 2 is root	local winner!
root trial #6: node 6 is root	local winner!
root trial #8: node 6 is root	

Low-overhead distribution of routing tables



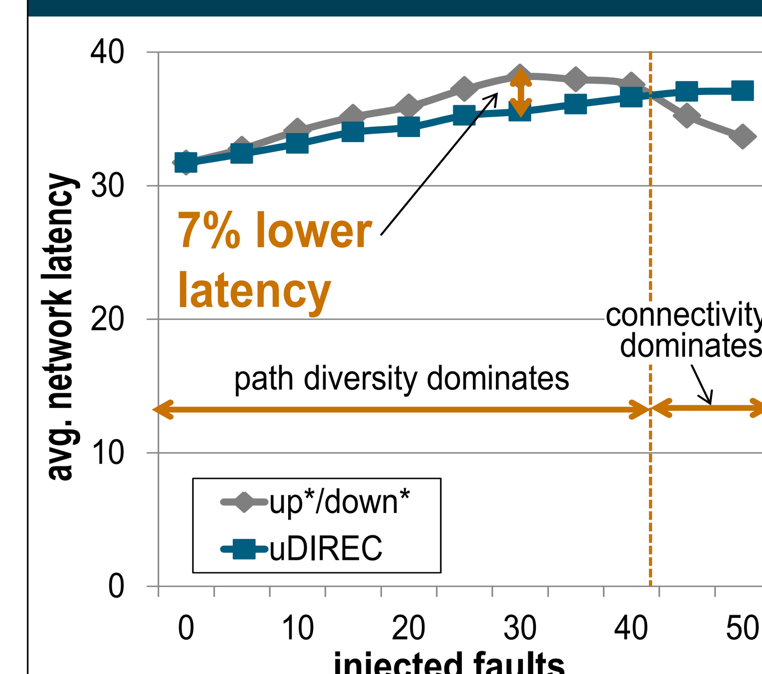
Reliability Results

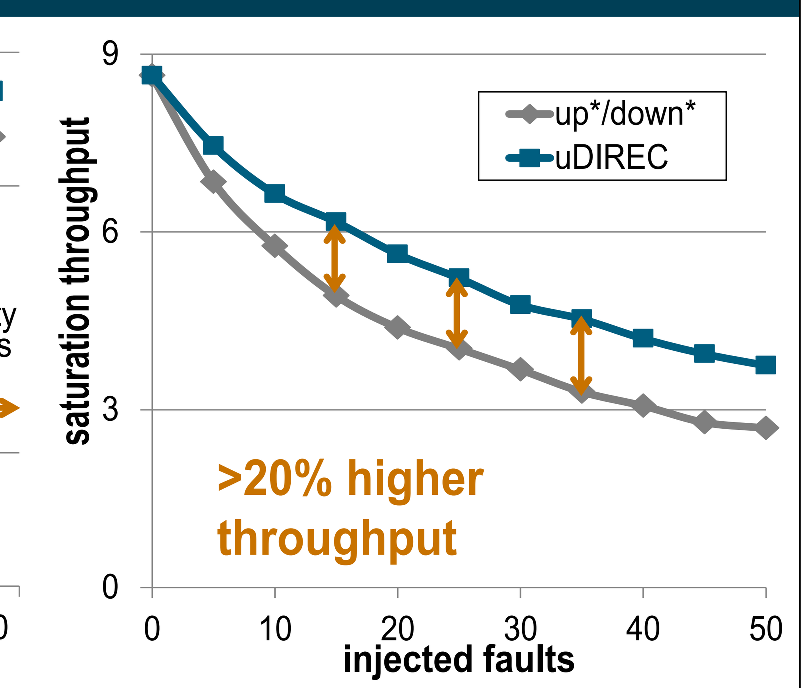




— **As faults accumulate networks become disconnected**
— **uDIREC looses fewer nodes and partitions into fewer networks**

Performance Results





— **Initially latency degrades gracefully; at higher fault rates up*/down* drops much more nodes, hence lower latency**
— **uDIREC consistently delivers higher throughput**