Application-aware Routing in Faulty Networks-on-Chip

Doowon Lee, Ritesh Parikh and Valeria Bertacco

Center for Future Architectures Research

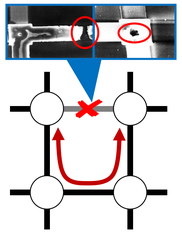
CFAR

STARnet

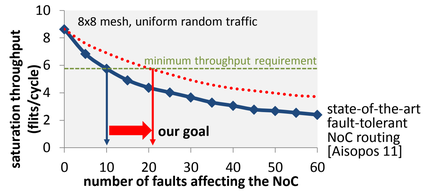
Theme 2384.005, Robust Design
Theme 2384.002, Communication

TASK 5.3, Ultra-low cost mechanisms for fault and variability tolerant architectures
TASK 2.1, Application-adaptive interconnect

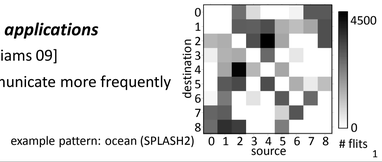
Faults Hurt…



permanent faults occur at runtime
➔ available network resources diminish
➔ performance degradation incurred due to congestion



distinct communication patterns in applications
multi-threaded workloads [Barrow-Williams 09]
➔ some source-destination pairs communicate more frequently

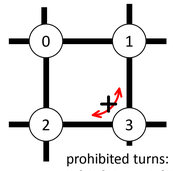


example pattern: ocean (SPLASH2)

1

Application-aware Adaptive Routing

deadlock-free adaptive routing by turn prohibition

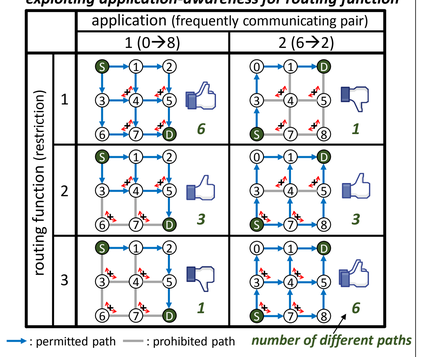


prohibited turns:
2➔3➔1 and 1➔3➔2

1. turn model
[Glass92, Wu03, Fu11]
− fault-tolerance
− application-awareness

2. topology-agnostic routing
[Aisopos11, Parikh13]
+ fault-tolerance
− application-awareness

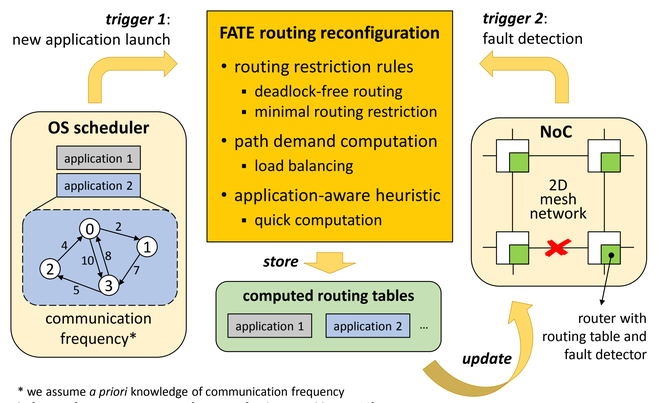
exploiting application-awareness for routing function



OUR GOAL: maximize network performance under many faults by **application-aware routing**

2

Fault- and Application-aware Turn-model Extension (FATE)



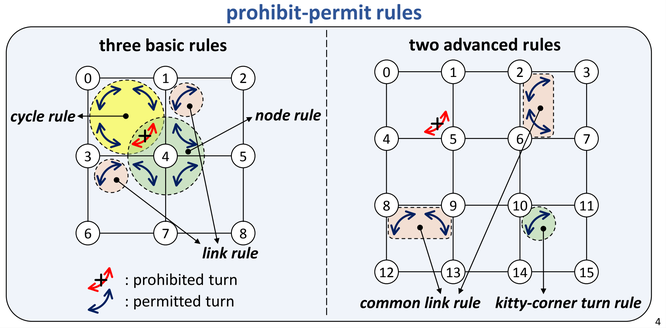
* we assume *a priori* knowledge of communication frequency before performing routing reconfiguration [Badr14, Waddington08].

3

Routing Restriction Placement Rules

- complexity of finding optimal deadlock-free routes is **NP-complete** [Festa99]
- quickly find **sub-optimal routes** using deadlock-free routing restriction rules
  - minimal routing restriction
  - deadlock-free rules tailored to 2D meshes

prohibit-permit rules

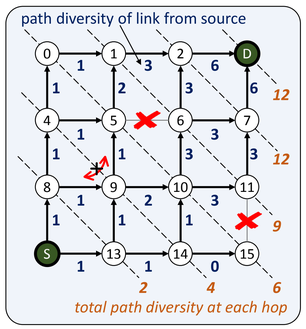


4

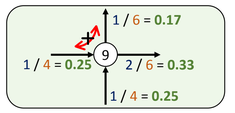
Path-Demand Computation

- mitigate congested situations via **path diversity**
- statically estimate the **demand of links and turns**

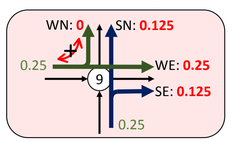
(1) **path diversity**



(2) **link demand** = path diversity of link / total path diversity

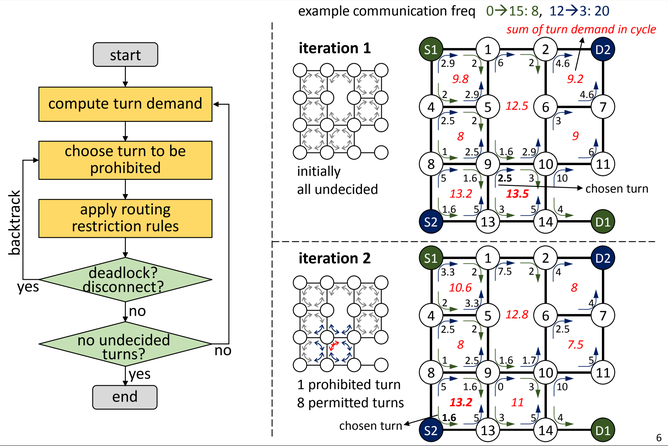


(3) **turn demand** = link demand / # permitted turns



5

Application-aware Routing Restriction Heuristic



6

Faulty Networks — Synthetic Traffic Patterns

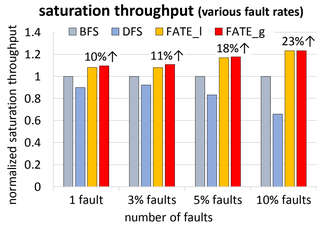
experimental setup

network	Booksim, 8×8 mesh
router	input-buffered, 3-stage, 5 flits/VC, 2 VCs/port

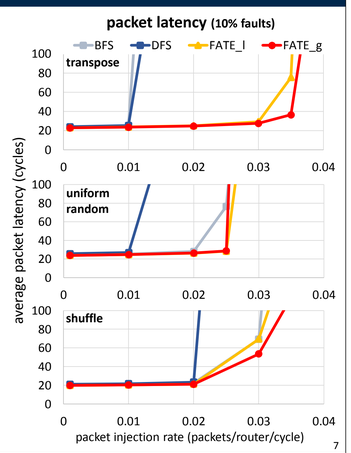
solutions compared

BFS	breadth-first search up*/down*	**FATE_l**	FATE with local output probability
DFS	depth-first search up*/down*	**FATE_g**	FATE with global path diversity

saturation throughput (various fault rates)



packet latency (10% faults)



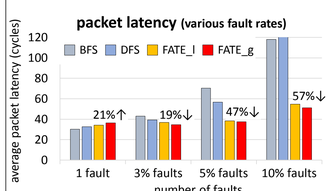
7

Faulty Networks — SPLASH2 Benchmarks

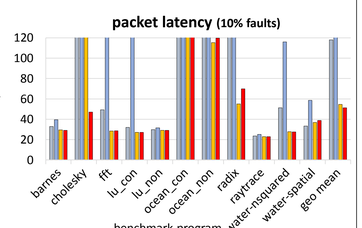
SPLASH2 trace configuration

simulator	gem5 syscall emulation mode	L1 cache	private, 32 KB
cache coherence	Ruby, MESI, CMP directory	L2 cache	shared, 256 KB
network	Garnet, 8×8 mesh	memory	4 GB total, 4 controllers at corners
		trace length	10 million cycles from cache init.

packet latency (various fault rates)



packet latency (10% faults)



conclusion — faulty networks
- **up to 22% improvement** in saturation throughput for synthetic traffic patterns
- **up to 57% improvement** in packet latency for SPLASH2 traces
- **graceful performance degradation with faults ↑**

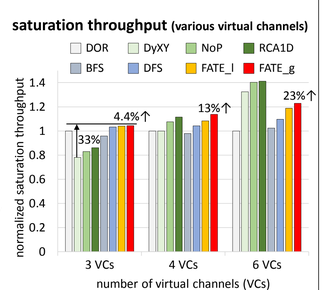
8

Non-faulty Networks — Synthetic Traffic Patterns

solutions compared

DOR	-	deterministic, X-Y dimension order routing	escape VC necessary for deadlock recovery
DyXY	[Li06]	fully adaptive, 1-hop congestion	
NoP	[Ascia08]	fully adaptive, 2-hop congestion	
RCA1D	[Gratz08]	fully adaptive, 1-dimensional congestion	
BFS	[Aisopos11]	partially adaptive, 1-hop congestion	escape VC unnecessary
DFS	[Sancho04]		
FATE_l	ours		
FATE_g			

saturation throughput (various virtual channels)



conclusion — non-faulty networks
- **up to 23% improvement** in saturation throughput over DOR
- **up to 33% improvement** in saturation throughput over fully-adaptive routing (higher benefit under fewer VCs)

9