

Guaranteeing Deadlock Freedom in Arbitrary Network Topologies using Packet Swaps

Mayank Parasar
(mparasar3@gatech.edu)
Graduate Student

Academic Advisor: Tushar Krishna
(tushar@ece.gatech.edu)
Assistant Professor

School of Electrical & Computer Engineering
Georgia Institute of Technology, Atlanta, GA

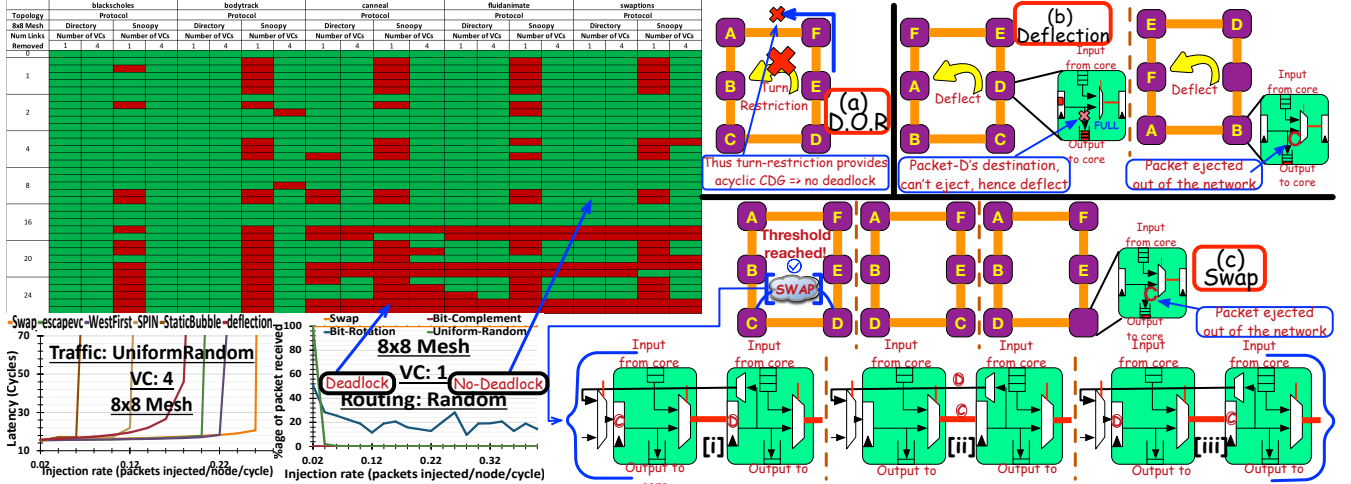


Fig. 1: Upper left of the figure shows the results from full system simulations ran on gem5 in which parsec-applications incur deadlock, for both snoopy and directory based cache coherence protocol. Underlying irregular topology is derived from 8x8 Mesh after disabling 1, 2, 4, 8, 16, 20 and 24 links respectively. Here red-color indicate deadlock and green-color indicate no-deadlock. Right side of the figure differentiate *Dimension Order Routing* (DOR), Deflection routing and Swap-technique by showing the network-state. Bottom left of the figure, left-graph shows swap-technique out-performs state-of-art deadlock-recovery (*spin*, *static_bubble*) and deadlock-avoidance (*west-first*) scheme. Bottom left of the figure, right-graph shows the correctness of swap-technique when applied to baseline random-routing, as all the packets are ejected from the network. However without swap-technique baseline routing incurs deadlock.

I. PROBLEM AND MOTIVATION

Interconnection networks are the communication backbone for any computing system. It allows exchange of messages/packets among different nodes of the system, be it an on-chip or an off-chip system. Different nodes could be a CPU-core, accelerators (such as GPUs, DNN-accelerator etc) or even a computer cluster connected with other different computer clusters, depending on what granularity we define 'node'. One of the most fundamental challenges in any interconnection network is that of routing deadlocks. A deadlock is a cyclic dependence between buffers that renders forward progress impossible.

Deadlocks are necessary evil and almost every on-chip/HPC network today avoids it either via routing restrictions across physical channel (Dally's theory) or with at least one escape virtual channel (Duato's Theory). This makes the *Channel Dependency Graph* (CDG) acyclic thus making sure that a cyclic dependence between buffers is never created in the first place. However, the analysis of making CDG acyclic is tied closely to the underlying network topology, which makes it difficult to port off-the-shelf deadlock free routing algorithm to irregular topologies. Moreover, irregular topologies more likely to form dynamically during the runtime of the system even if we begin from a regular topology such as Mesh. This is due to the power gating and dynamic link failures. This can make the earlier deadlock-free routing to deadlock in the new topology which is dynamically formed. We study how often deadlock occur in a full system simulation on gem5 [1] with garnet2.0 [2] network model, with parsec benchmark [3] running on latest Linux kernel.

Therefore, with cyclic-CDG it is not correct to ask, "if deadlock will occur?", instead the right question is "when deadlock will occur?".

II. BACKGROUND

The problem of deadlocks has received significant attention from research community and we broadly classify the theoretical framework for deadlock freedom in 5 categories:

- 1) Dally's theory [4] defines a strict order in acquisition of links and/or buffer resources/Virtual Channels (VCs) by network packets which ensures that a cyclic dependence is never created, as shown in the top figure.

- 2) Duato's theory [5] introduces the idea of escape paths that packets in a cyclic dependence can use to avoid from deadlocks.
- 3) Flow control based schemes [6], [7] prevent packet injection, when the number of empty buffers in the network reach certain minimum to ensure the presence of at least one free buffer in dependency chain to ensure forward progress.
- 4) Deadlock Recovery [8], [9] based schemes, argue that instead of allocating resources to prevent deadlock to occur, one should detect it and then recovers from it. This involves extensive control circuitry to detect the deadlock and then recovers from it.
- 5) Deflection Routing [10], [11], [12], [13] schemes deflect/mis-route the packets to other outports, if more than one packet requests to go through same outport.

III. APPROACH AND NOVELTY

We introduce the concept of *inplace-packet-swaps* between adjacent routers as a means to provide deadlock freedom. In the context of this proposal, swap refers to the act of exchanging two packets, between two routers, *such that at least one of the packet (involved in swap-operation), makes forward progress*. We randomly choose the inport of the router, and swap its packet with corresponding neighbor router. We prove that a finite bounded number of packet swaps are sufficient to break any deadlock.

Nuance about swap-operation, is that there is no credit management needed between upstream and downstream routers, because there is no change in overall occupancy of buffers. We highlight that our proposal is fundamentally different from earlier proposed solution, especially deflection routing in following ways:

- 1) It does not provide any turn restrictions that packet can take, unlike deadlock avoidance (Duato and Dally's theory) hence it provides full path diversity.
- 2) Swap technique chooses a unique router every-time to perform swap based on router-id (remember necessary condition for swap necessitates both routers to have packets).
- 3) Unlike deflection routing causing network-wide deflection, in Swap only chosen router selectively performs the swap.

IV. CONTRIBUTION

Following are the contribution this work:

- 1) Swap technique does not incur the overhead of deadlock-detection or flow control scheme
- 2) Unlike Duato's or Dally's theory it provides full path-diversity.
- 3) Unlike Deflection routing, it provides the knob to control amount of deflection in network.
- 4) It works with any arbitrary/irregular topology that may occur at design time due to heterogeneous IPs or at runtime due to faults or power gating.

V. RESULTS:

Fig. 1 left graphs show (a) Performance of "Swap" as it provides more path-diversity and out-perform state-of-art deadlock recovery scheme by $2.17\times$, deadlock-avoidance scheme by 20% and deflection routing by 30% on average. (b) Correctness of "Swap", that it does not deadlock in any synthetic traffic pattern [14] and all injected packets eventually eject out of the network.

REFERENCES

- [1] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.
- [2] N. Agarwal *et al.*, "GARNET: A detailed on-chip network model inside a full-system simulator," in *ISPASS*, 2009.
- [3] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '08. New York, NY, USA: ACM, 2008, pp. 72–81.
- [4] W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Comput.*, pp. 547–553, 1987.
- [5] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Trans. Parallel Distrib. Syst.*, 1993.
- [6] C. Carrion, *et al.*, "A flow control mechanism to avoid message deadlock in k-ary n-cube networks," in *HIPC*, 1997.
- [7] V. Puente *et al.*, "The adaptive bubble router," *J. Parallel Distrib. Comput.*, pp. 1180–1208, 2001.
- [8] A. Ramrakhiani and T. Krishna, "Static bubble: A framework for deadlock-free irregular on-chip topologies," in *HPCA*, 2017, pp. 253–264.
- [9] K. V. Anjan and T. M. Pinkston, "An efficient, fully adaptive deadlock recovery scheme: DISHA," in *ISCA*, 1995.
- [10] S. Konstantinidou and L. Snyder, "Chaos router: Architecture and performance," in *Proceedings of the 18th Annual International Symposium on Computer Architecture*, ser. ISCA '91. New York, NY, USA: ACM, 1991, pp. 212–221. [Online]. Available: <http://doi.acm.org/10.1145/115952.115974>
- [11] C. Fallin *et al.*, "Chipper: A low-complexity bufferless deflection router," in *HPCA*, 2011, pp. 144–155.
- [12] G. Michelogiannakis *et al.*, "Evaluating bufferless flow control for on-chip networks," in *NOCs*. IEEE Computer Society, 2010, pp. 9–16.
- [13] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," in *ISCA*, 2009.
- [14] "Garnet synthetic traffic." gem5.org. [Online]. Available: http://www.gem5.org/Garnet_Synthetic_Traffic