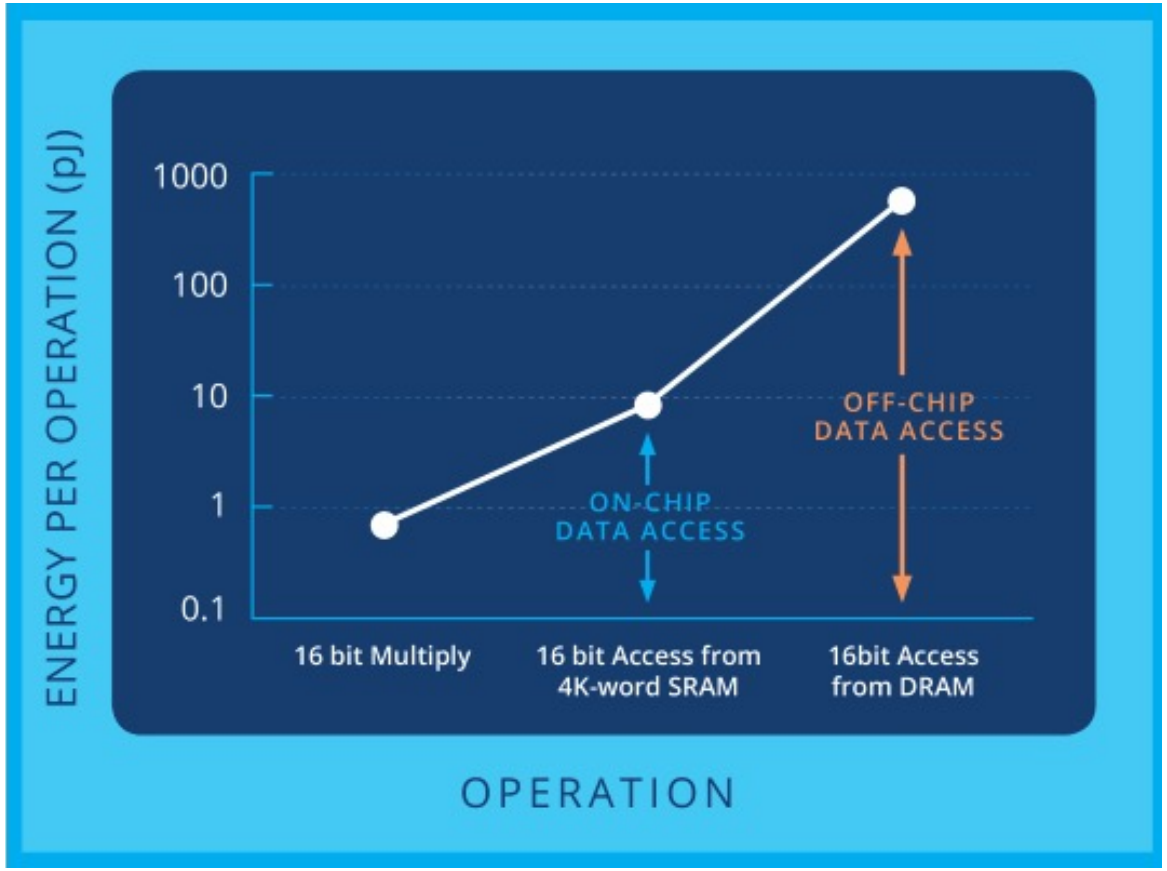
Communication bottleneck

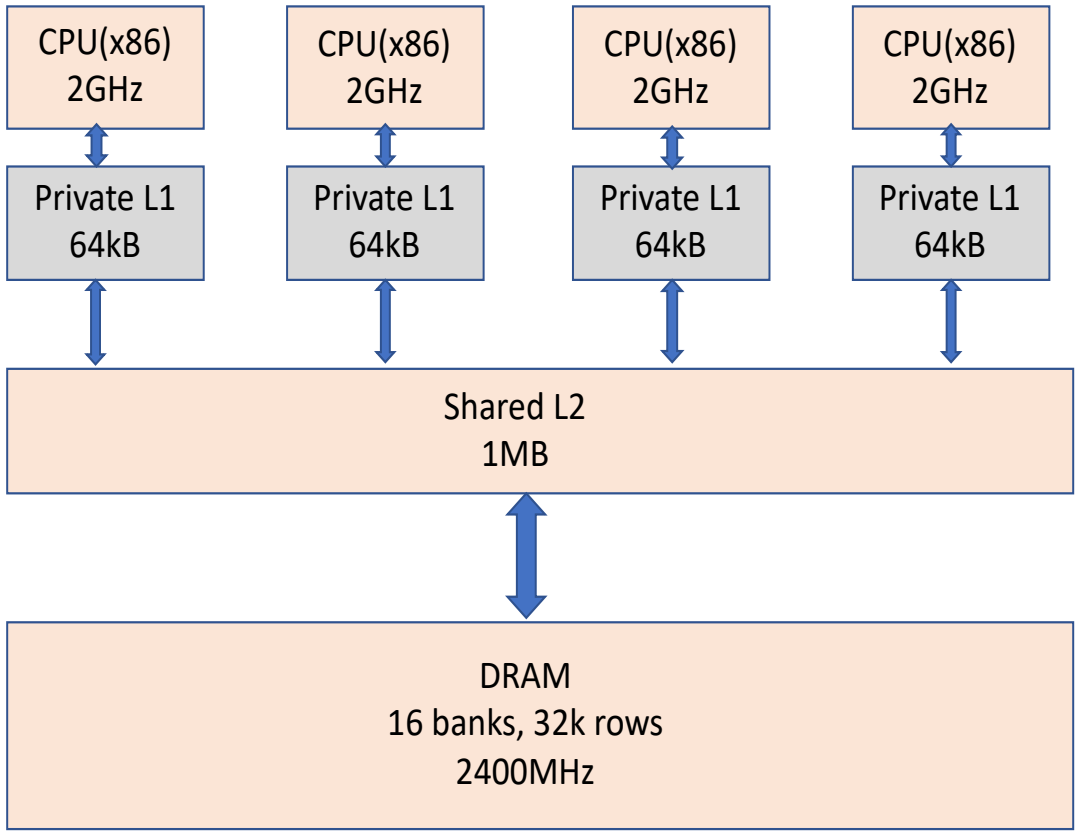


- Von Neuman architecture
- Computation is super fast!
- Communication is the new bottleneck

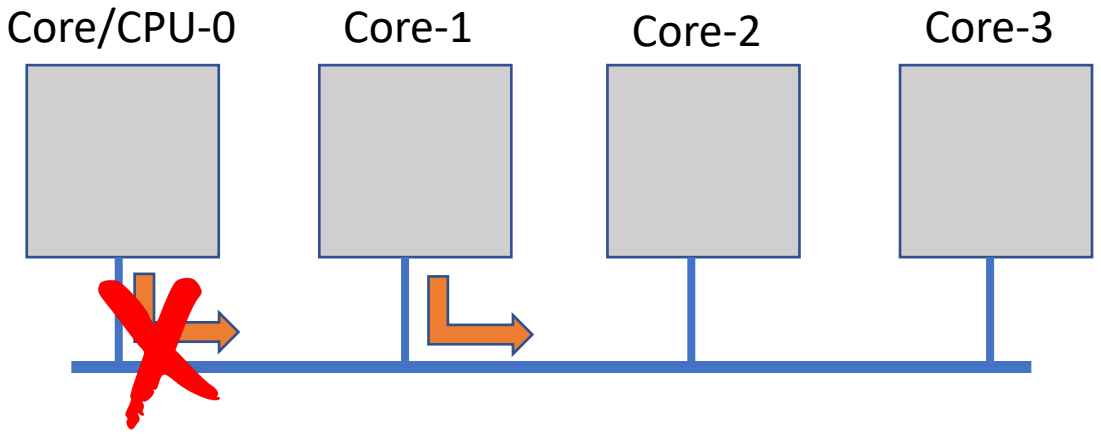
https://www.eetimes.com/the-race-is-on/#

Avoiding Memory bottleneck

- L2 to DRAM communication is slow
- DRAM is off-chip
- Will discuss in another lecture



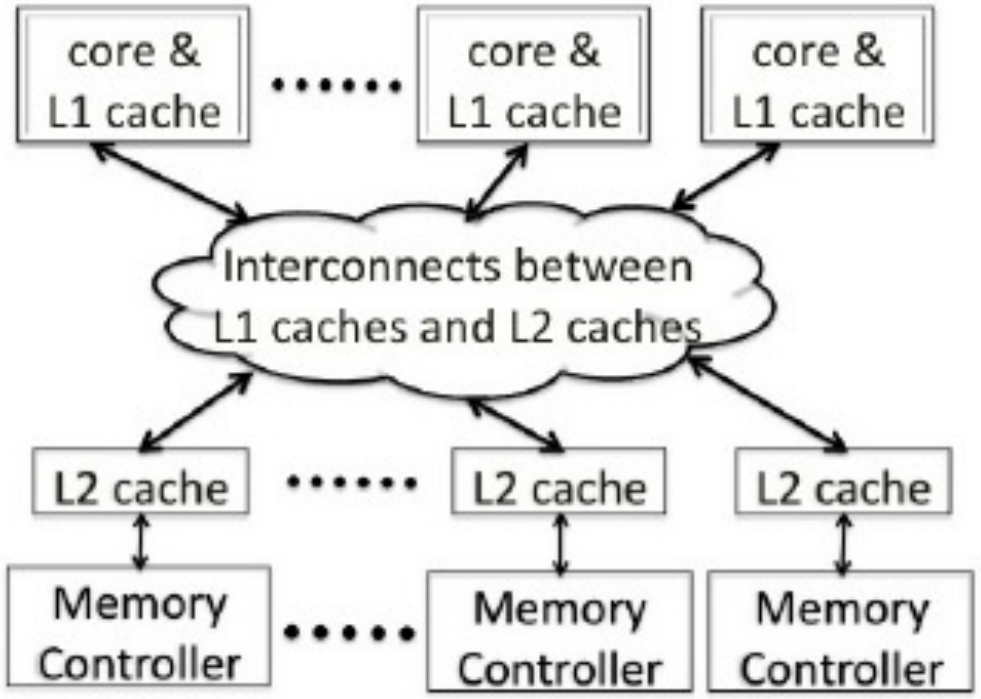
Communication in multicore



- **Shared communication bus**
- **Not scalable with core count**

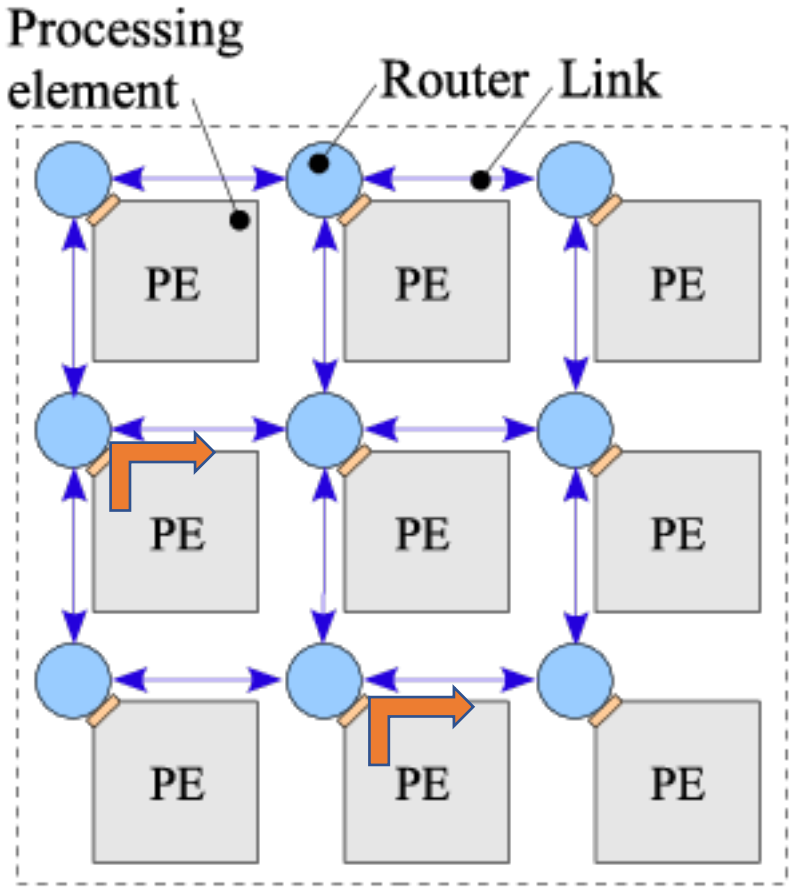
Solving communication bottleneck

- Shared medium of communication is not sufficient
- Some other interconnect infrastructure required
- Connect CPU, L2, and other components

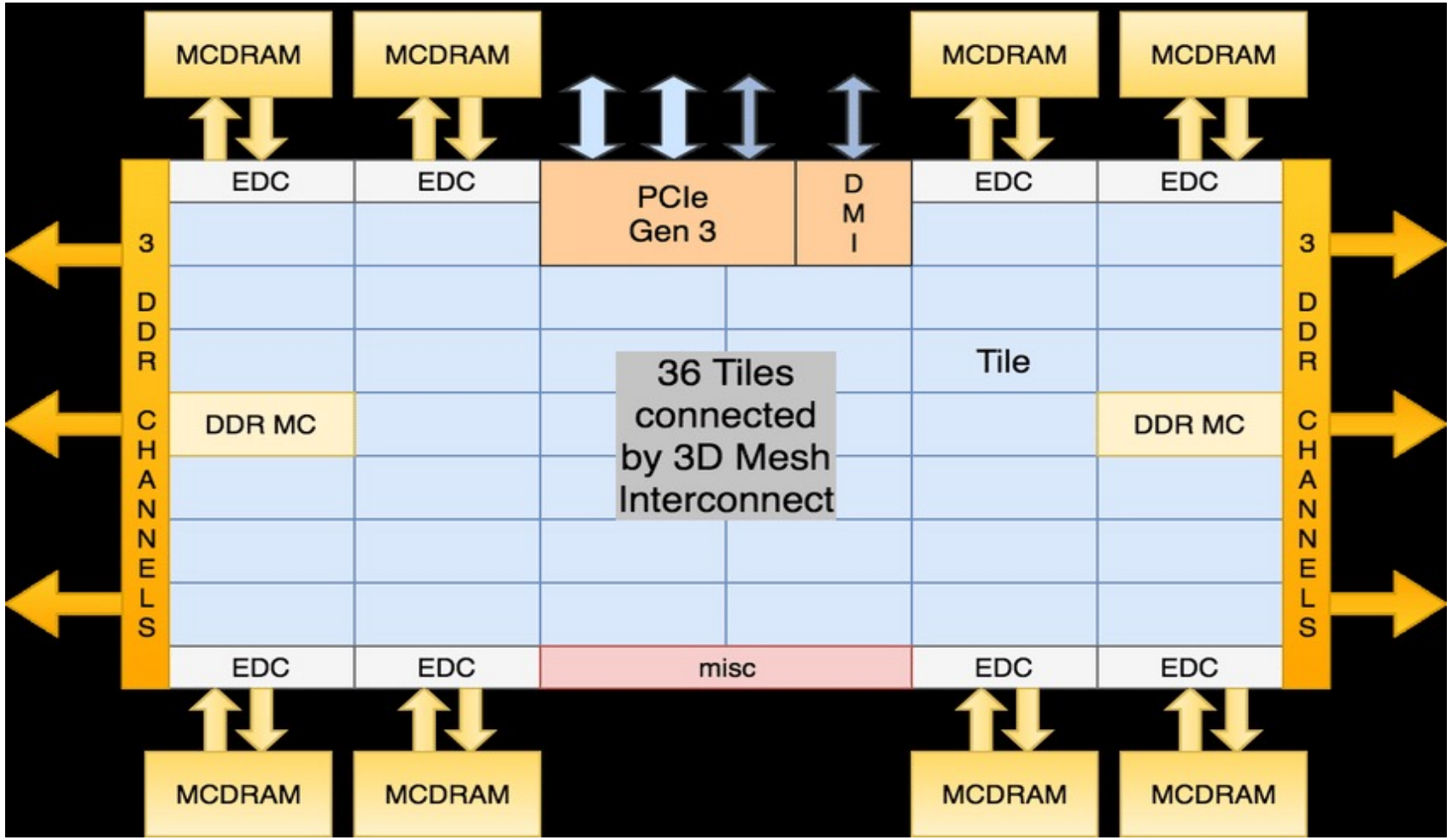


Network-on-Chip (NoC)

- Network-on-chip (NoC)
  - High throughput
  - Low latency
  - Energy efficiency
  - Scalable design

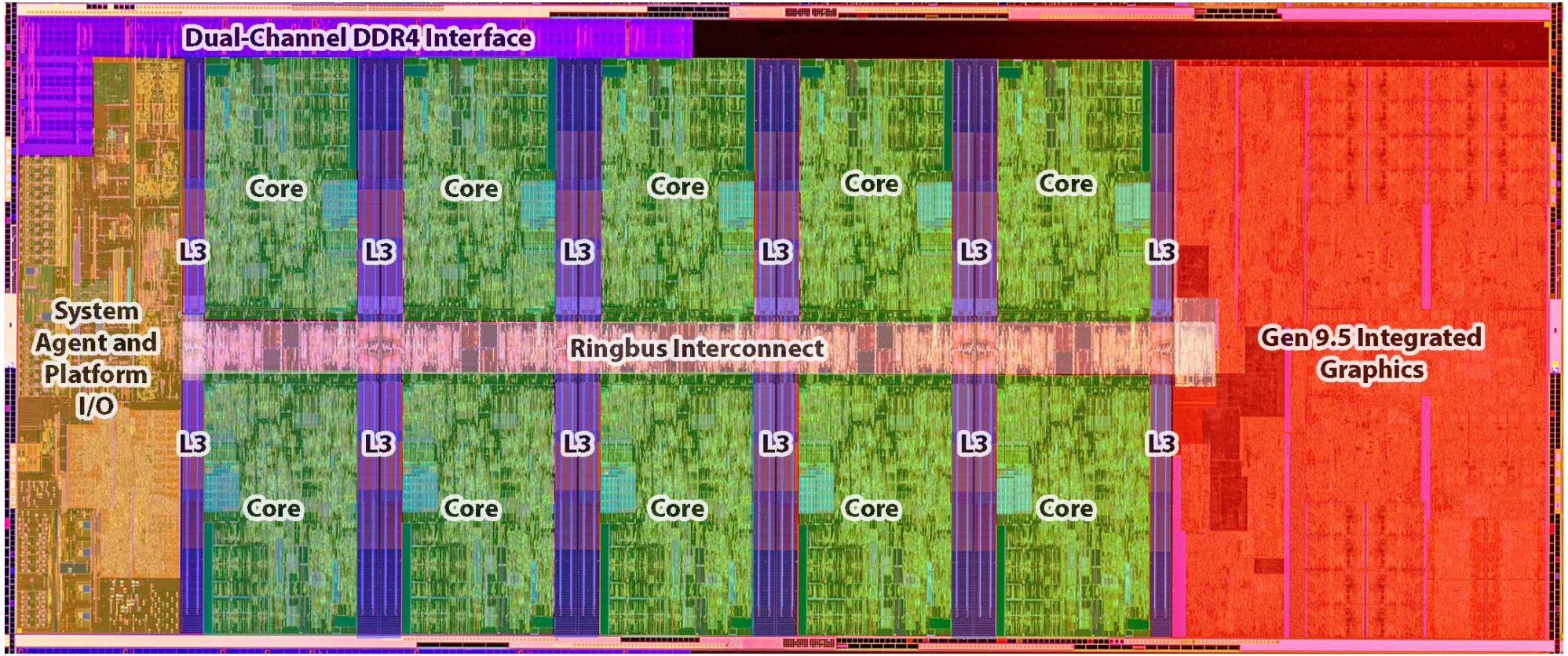


Network-on-Chip: Intel Xeon Phi Knights Landing



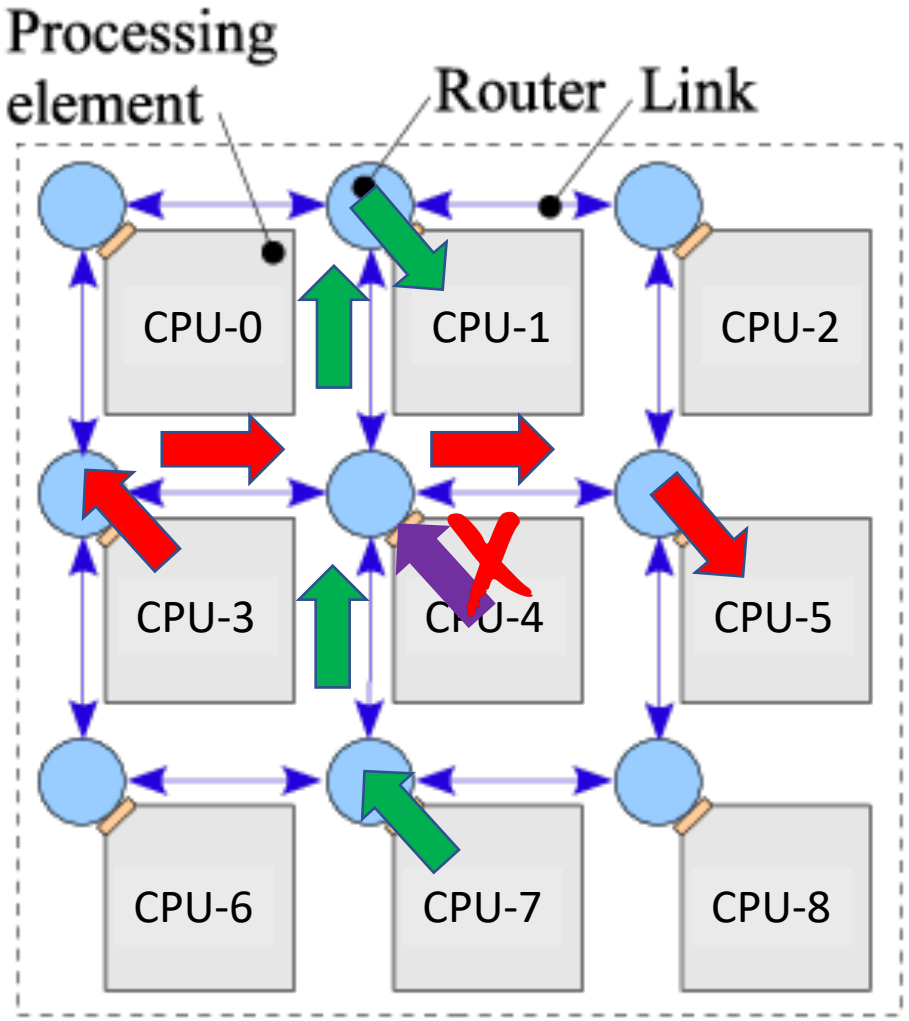
- **Mesh interconnect**

Network-on-Chip: Intel Core i9-10900K



- Ring interconnect

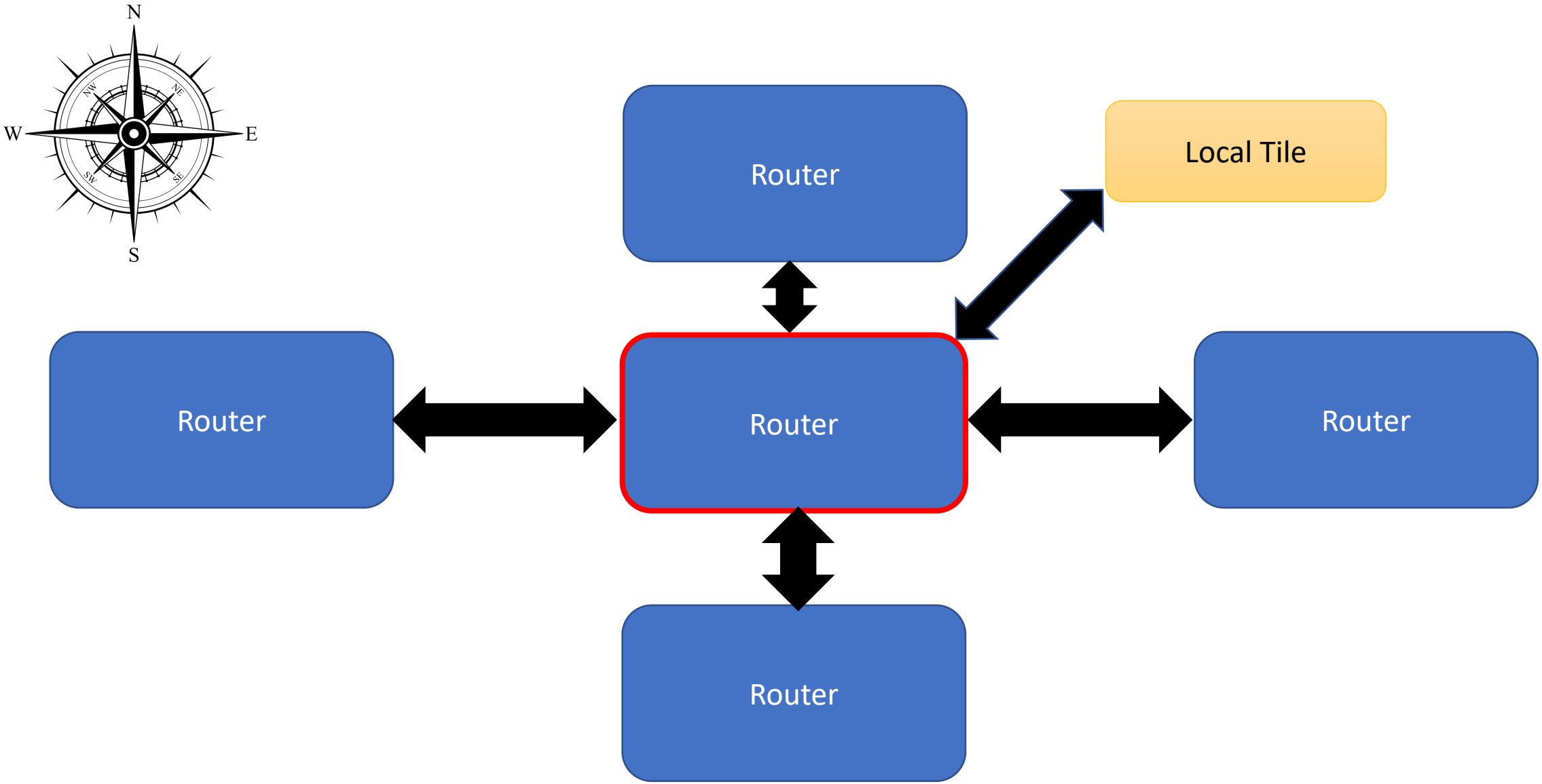
Communicating via the NoC



- CPU-3 to CPU-5
- Shortest path routing
  - Dijkstra's algorithm
  - Other routing exists
- CPU-7 to CPU-1
  - Can happen in parallel
- CPU-4 to CPU-5
  - Cannot happen
  - Link is busy

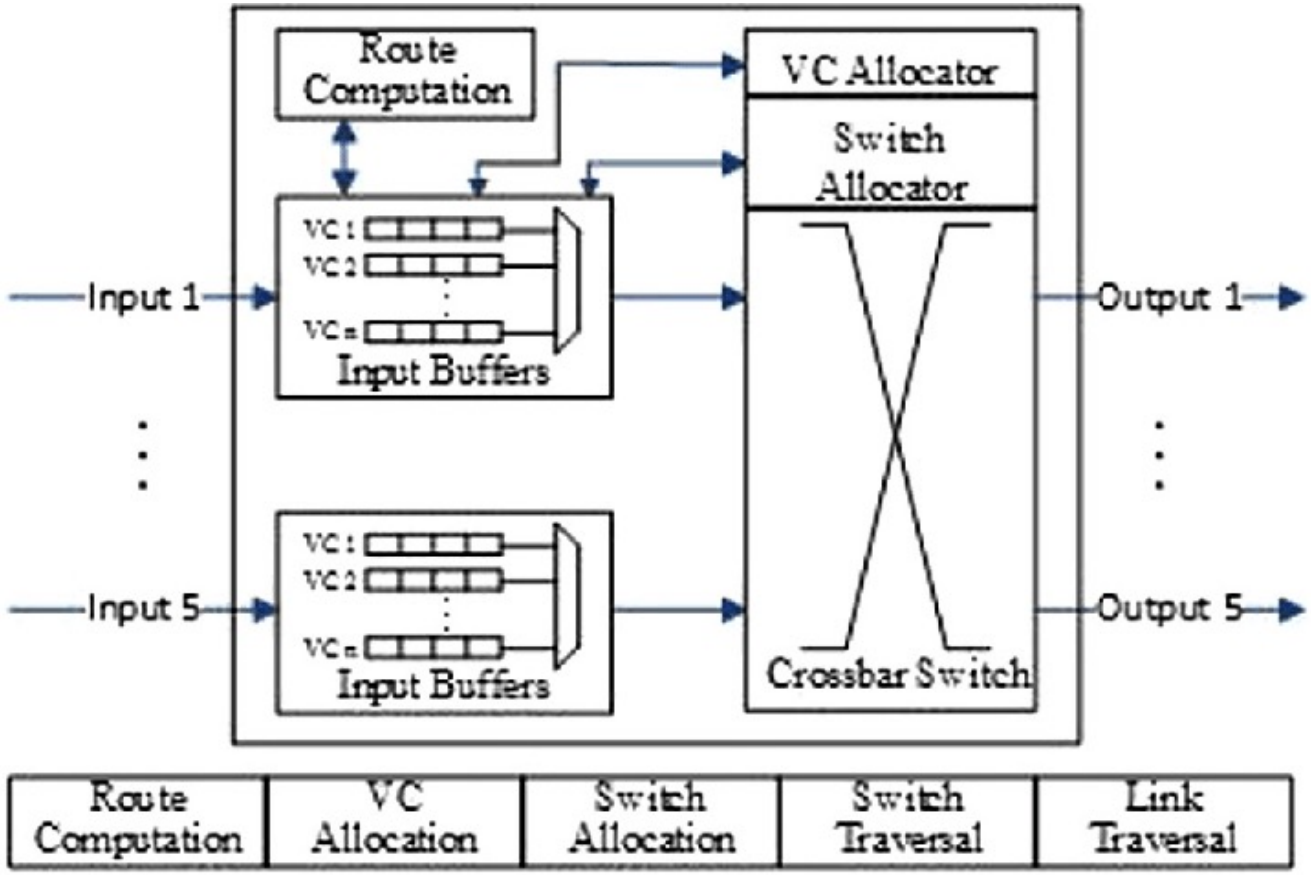
** will be in HW

NoC router (1)


N
W
E
S
NW
NE
SW
SE
Router
Local Tile
Router
Router
Router
Router

NoC router: Architecture

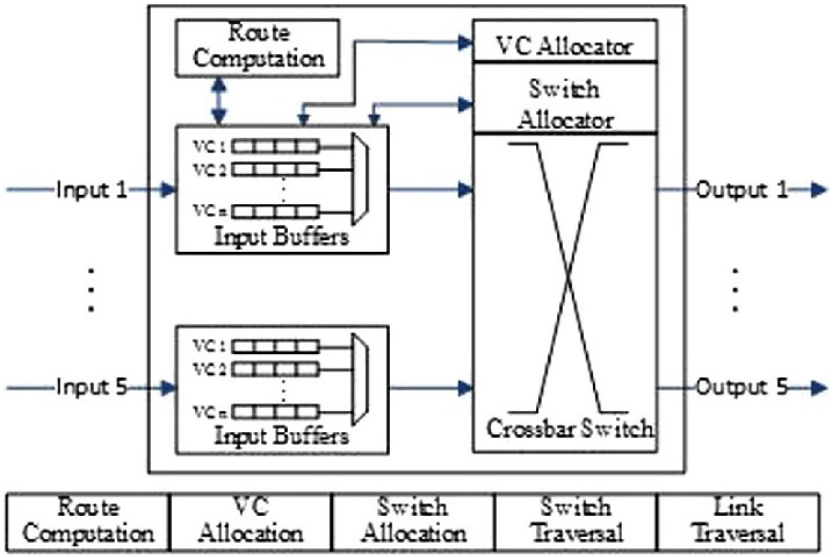
- 2D-Router usually has 5 ports
- Each port has buffers
- 5-stage pipeilne
  - Newer routers have fewer stages e.g., 1 or 3



Stages:

[1] A Multi-Synchronous Bi-Directional NoC (MBiNoC) architecture with dynamic self-reconfigurable channel for the GALS infrastructure
[2] SMART: A single-cycle reconfigurable NoC for SoC applications
[3] Tilera's iMesh, Intel's Ring, NoC prototypes (Park et al., DAC 2012)

NoC router: Details



(1) Route computation: this stage determines the output port that a packet must be sent to.

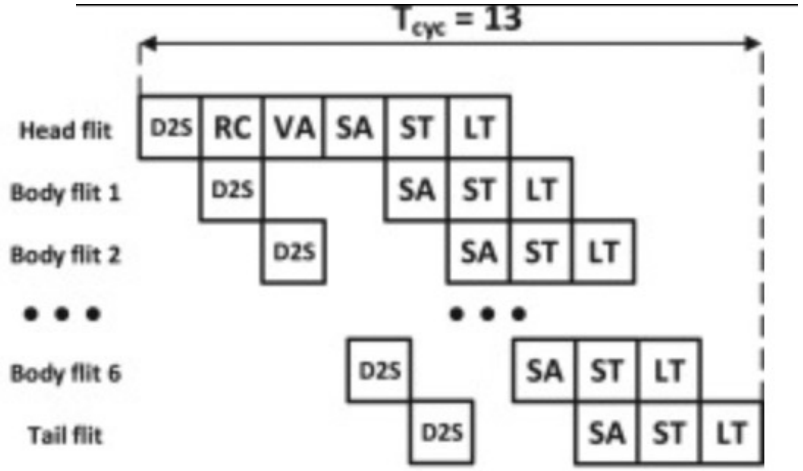
(2) VC allocation: this stage assigns an empty VC in the neighboring router connected to the output port. Since several header flits may send requests for the same VC, arbitration is required. The routing computation as well as the VC allocation only requires the header flit. The body and tail flits will follow their respective header flit.

(3) Switch allocation: if VC allocation is successful, the third stage sends request to the switch allocator to allocate the output port.

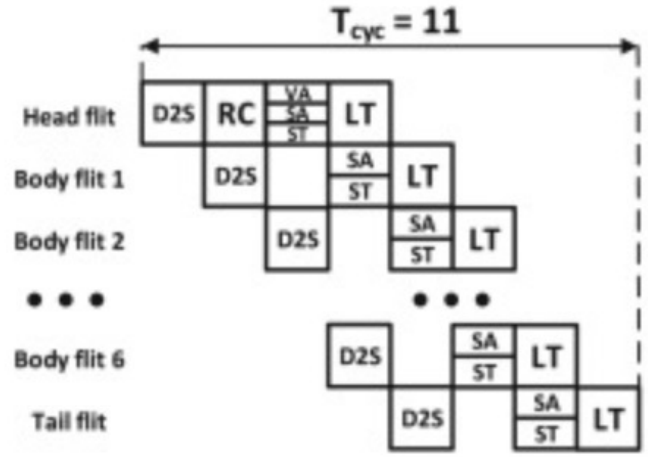
(4) Switch traversal: if the switch allocation is successful, the flit will be passed to the crossbar and be delivered to the output port.

Newer router architectures

- Newer router architectures require fewer cycles
- 3-cycle is common
- Up to 1-cycle router exist



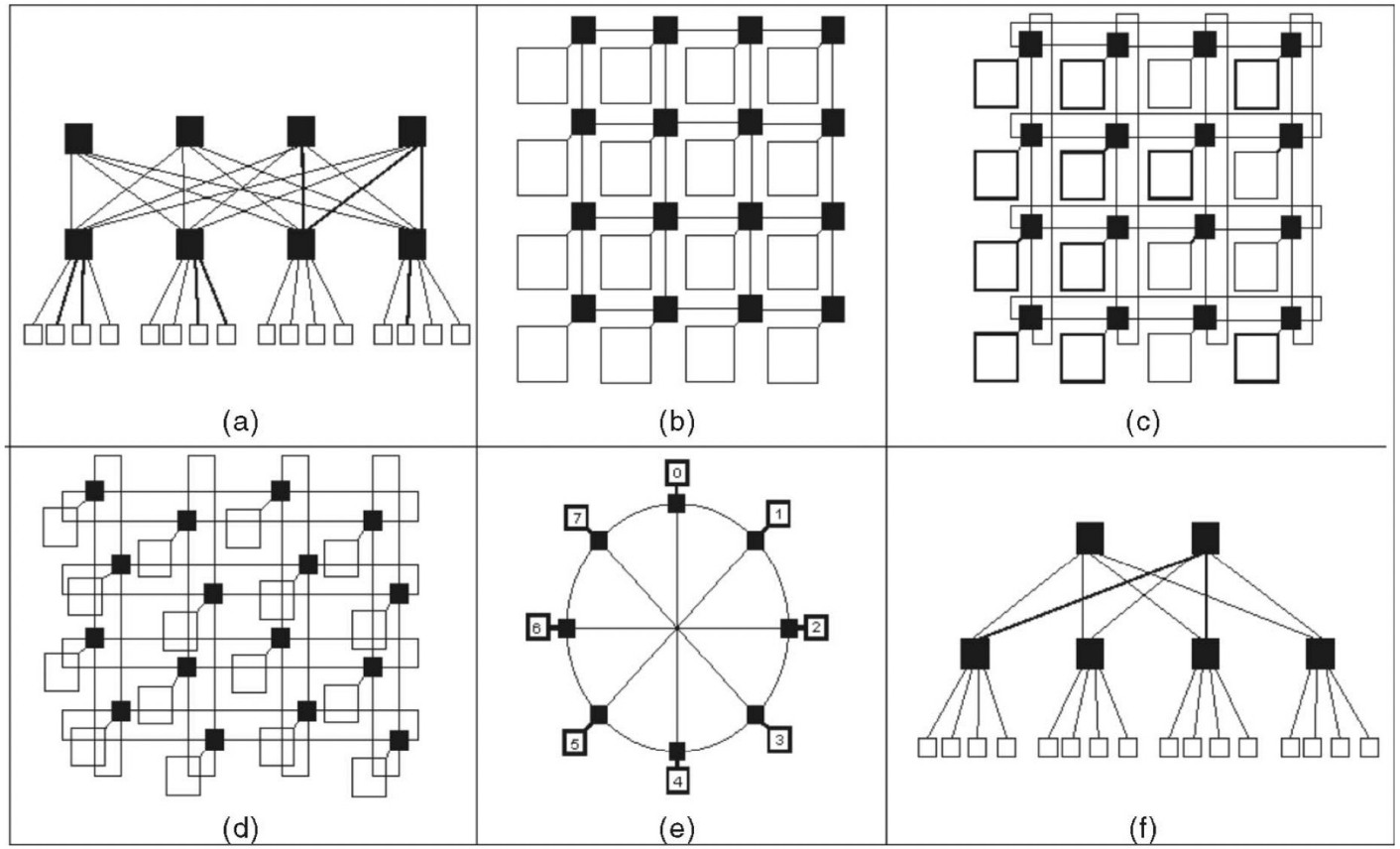
5-cycle PA-BiLink NoC router



3-cycle PA-BiLink router

J. Zhu et. al., "BiLink: A high performance NoC router architecture using bi-directional link with double data rate" in J. of Integration Volume 55, September 2016, Pages 30-42

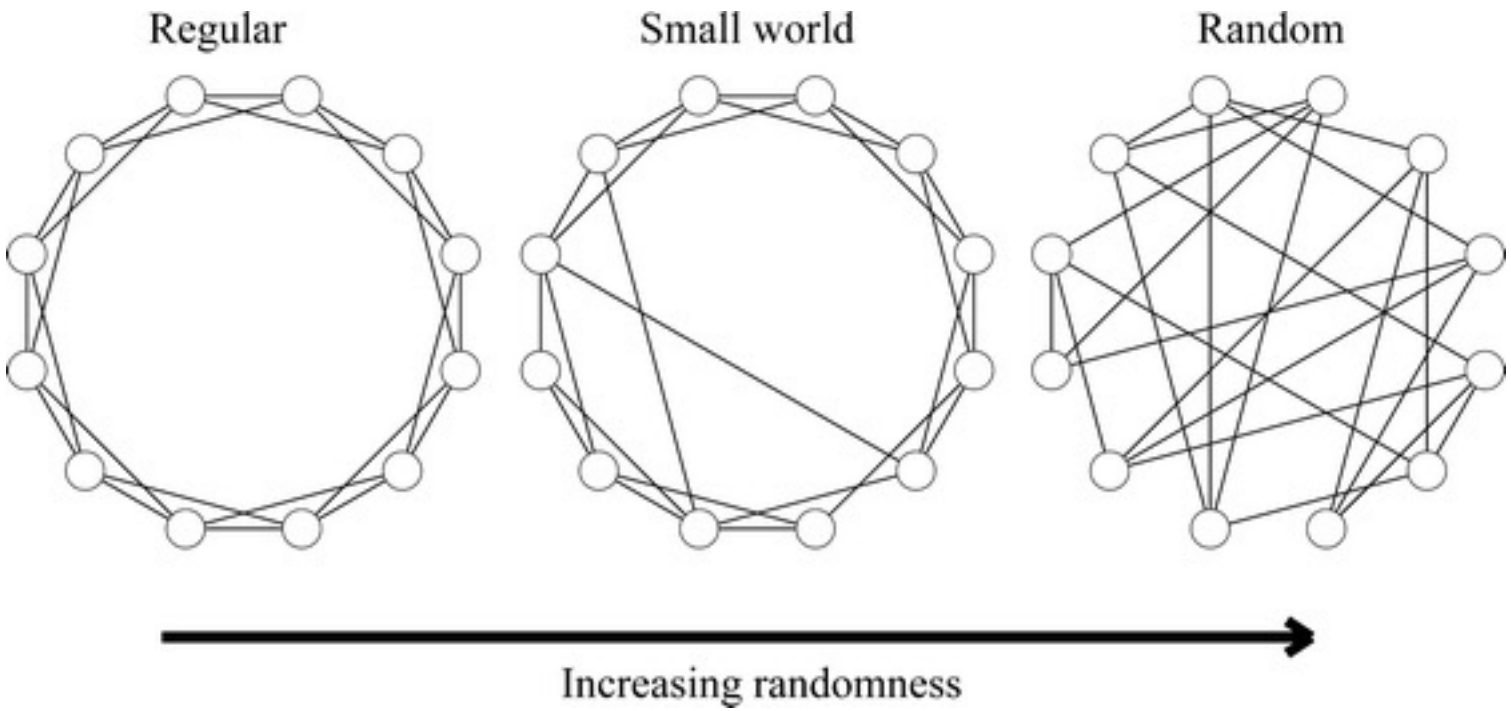
NoC configurations



- Many different NoCs
- Choice depends on designer
- Intels Knight's landing
  - Mesh NoC

P. P. Pande, C. Grecu, M. Jones, A. Ivanov and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," in IEEE Transactions on Computers, vol. 54, no. 8, pp. 1025-1040, Aug. 2005, doi: 10.1109/TC.2005.134.

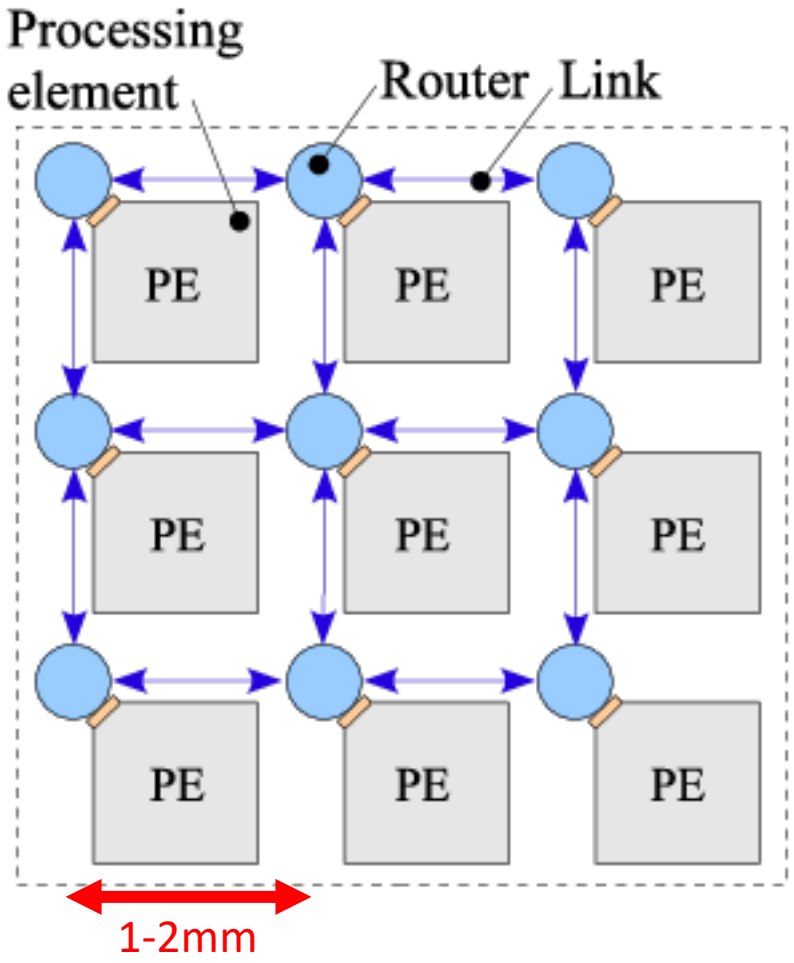
Small-world NoC

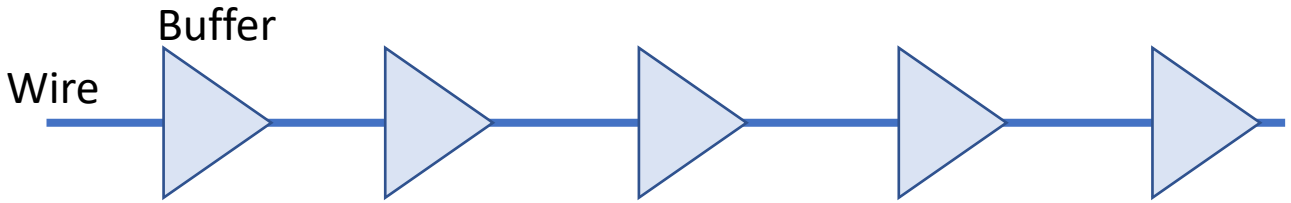


- **Regular NoC like Mesh are often sub-optimal**
- **Few long-range interconnect**
  - **Better power, performance**

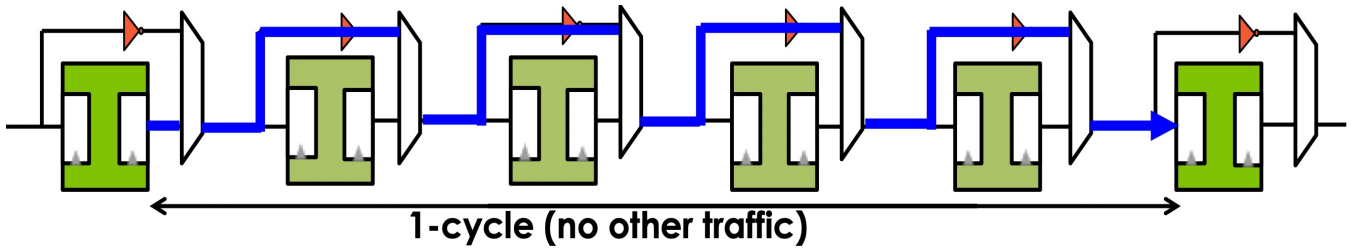
U. Y. Ogras and R. Marculescu, "It's a small world after all": NoC performance optimization via long-range link insertion," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 7, pp. 693-706, July 2006

SMART NoC



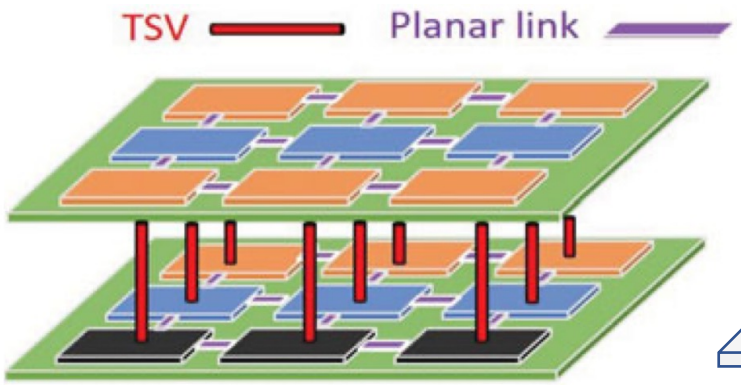


- Signal can travel 10-16mm in 1ns (1GHz)
  - Appropriate buffer sizing and placement
- SMART (Single-cycle Multi-hop Asynchronous Repeated Traversal)
  - Low Latency

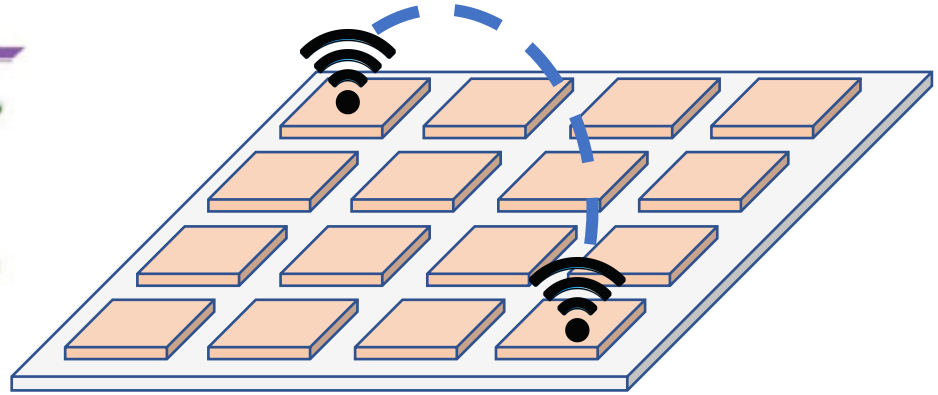


T. Krishna et. al., “SMART: A single-cycle reconfigurable NoC for SoC applications

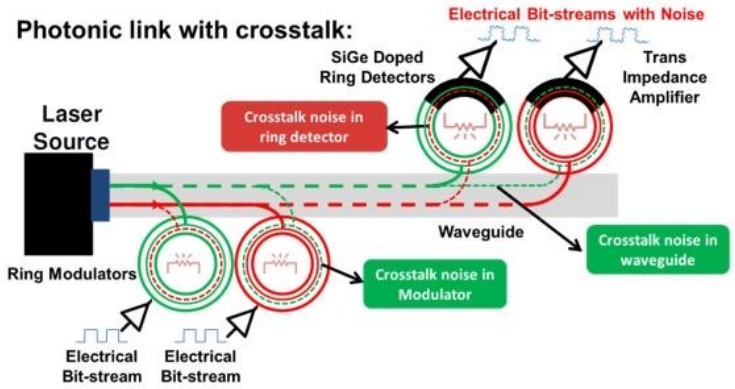
NoC technologies



3D NoC



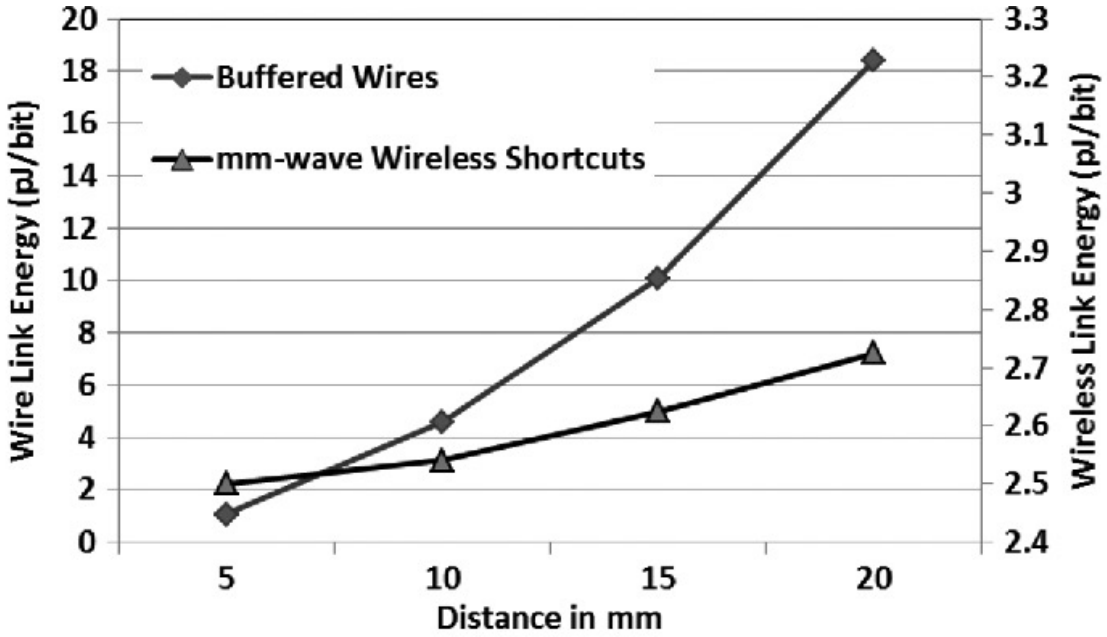
Wireless NoC

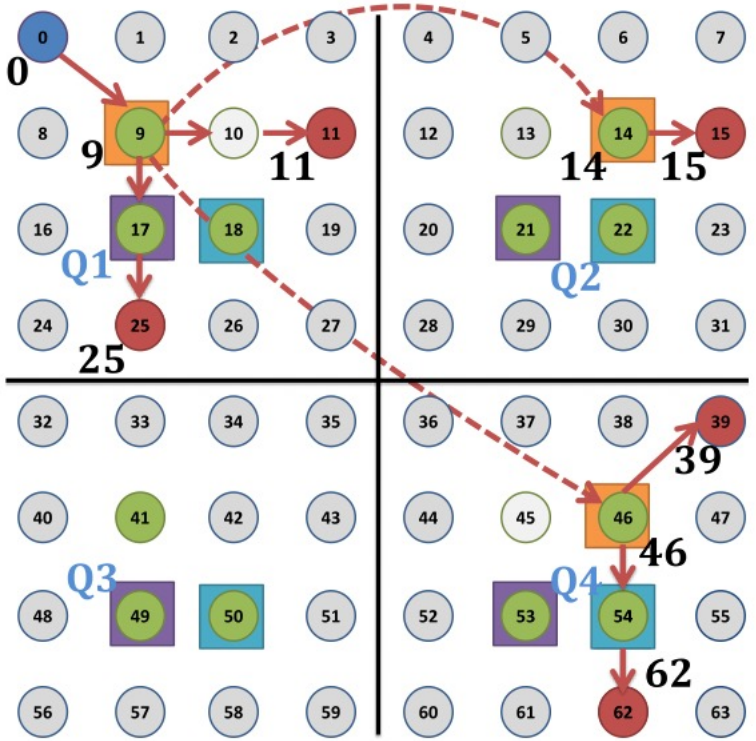


Photonics NoC

- **Different NoC technologies**
  - **Different pros and cons**

Wireless NoC

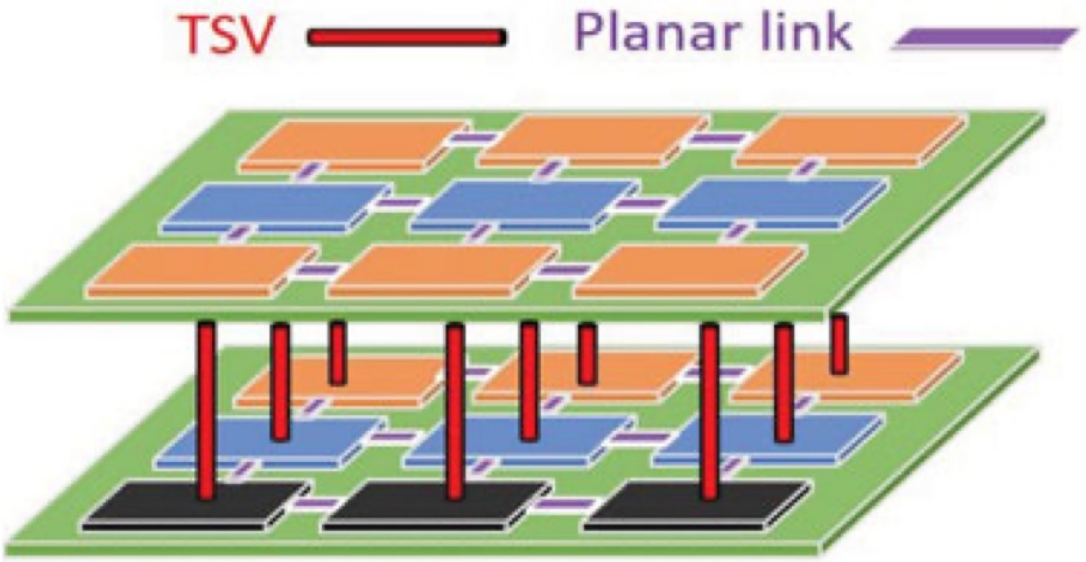




- **Wireless NoC**
  - **1-hop communication**
  - **Low bandwidth**
  - **Energy efficient for long-range communication**
  - **Good for multicast and broadcast traffic**

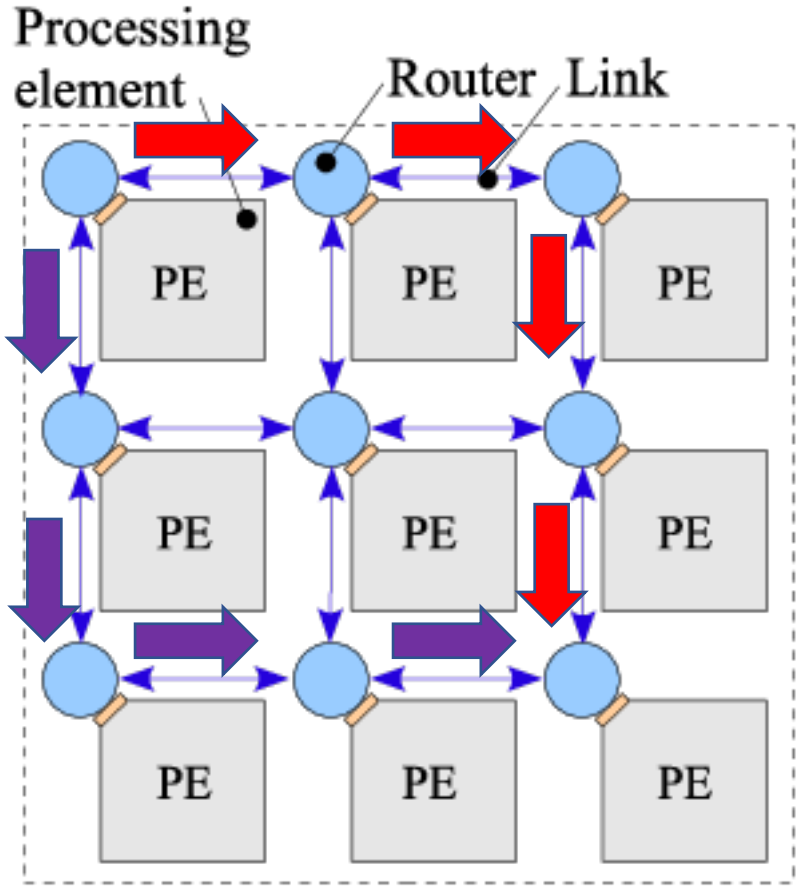
K. Duraisamy, Y. Xue, P. Bogdan and P. P. Pande, "Multicast-Aware High-Performance Wireless Network-on-Chip Architectures," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 3, pp. 1126-1139, March 2017, doi: 10.1109/TVLSI.2016.2612647.

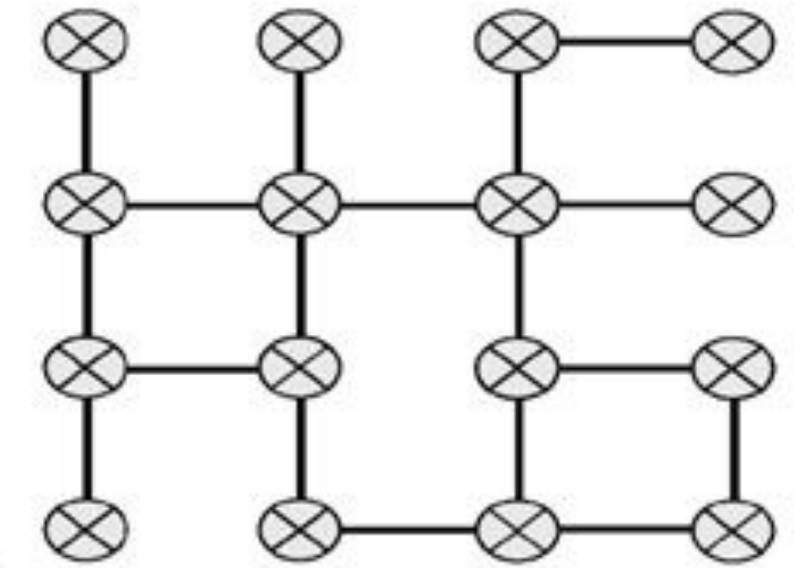
3D NoC



- 3D integration using TSV/M3D
  - Vertical links act as shortcuts
  - Denser integration, less footprint
  - More links than planar
    - Better performance
    - Has problems with heating

NoC Routing

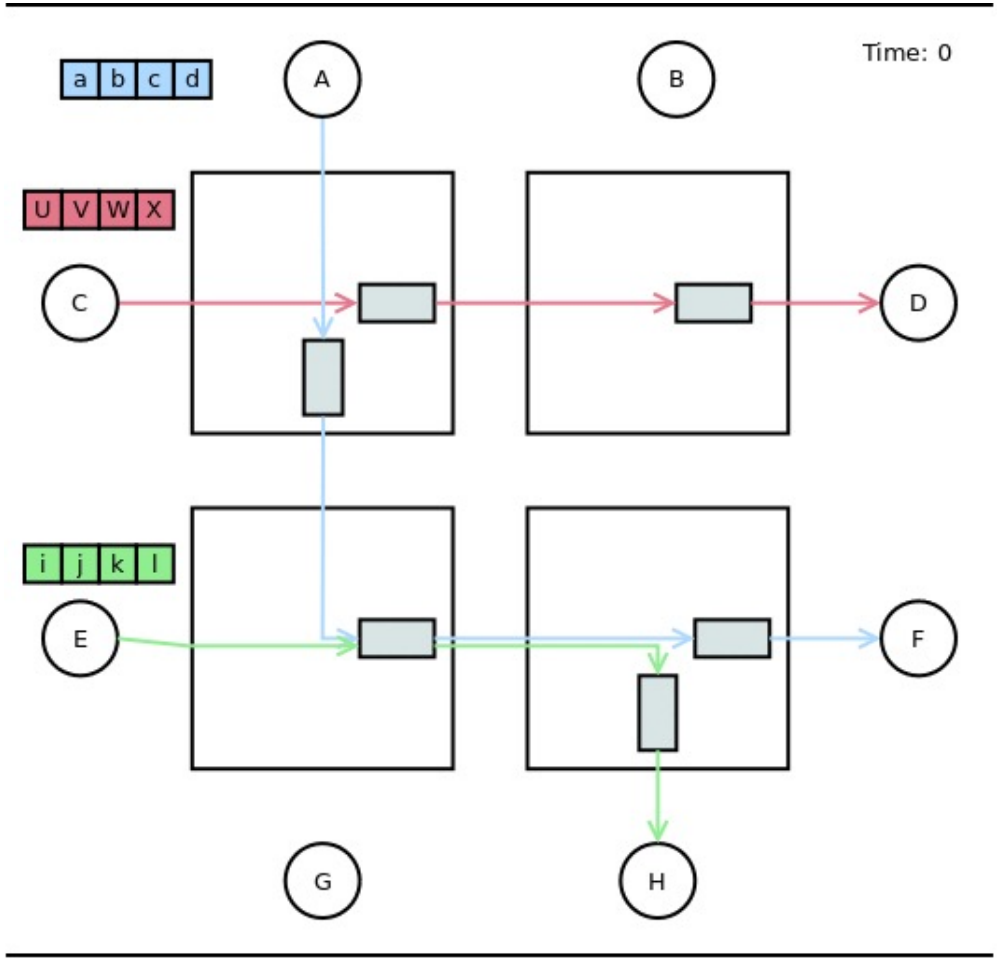




- Mesh NoC
  - XY and/or YX routing
  - Other routing exists
- Irregular NoC
  - Shortest path algorithm
  - Other routing exists

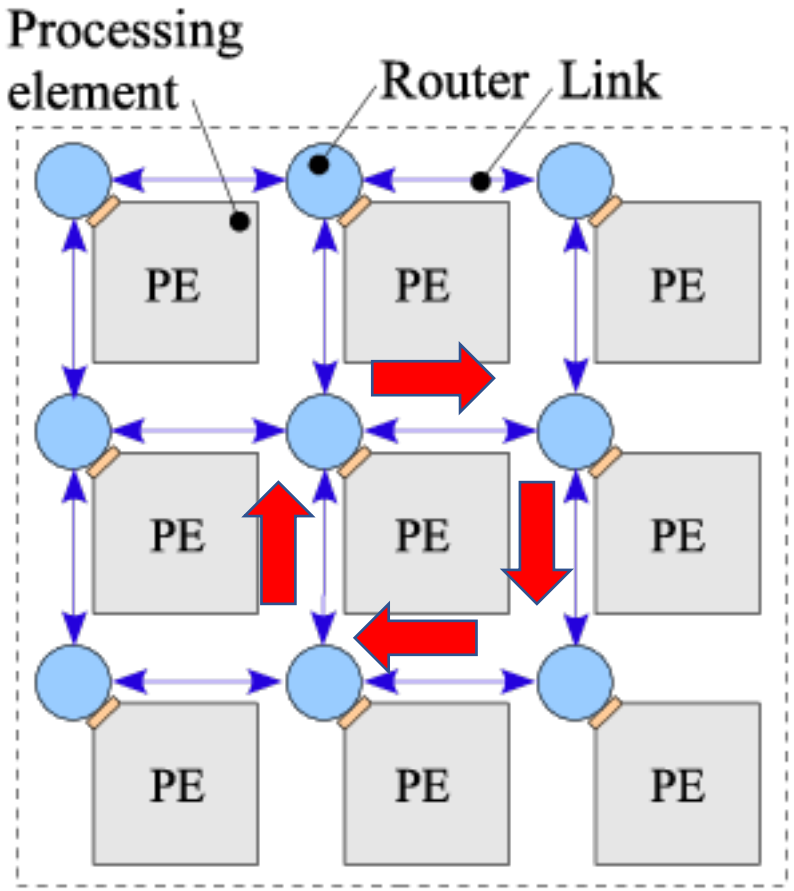
Wormhole routing

- Multi-flit messages
  - Sent in smaller pieces
- Header reserves a path
  - Body and tail flits follow the head flit



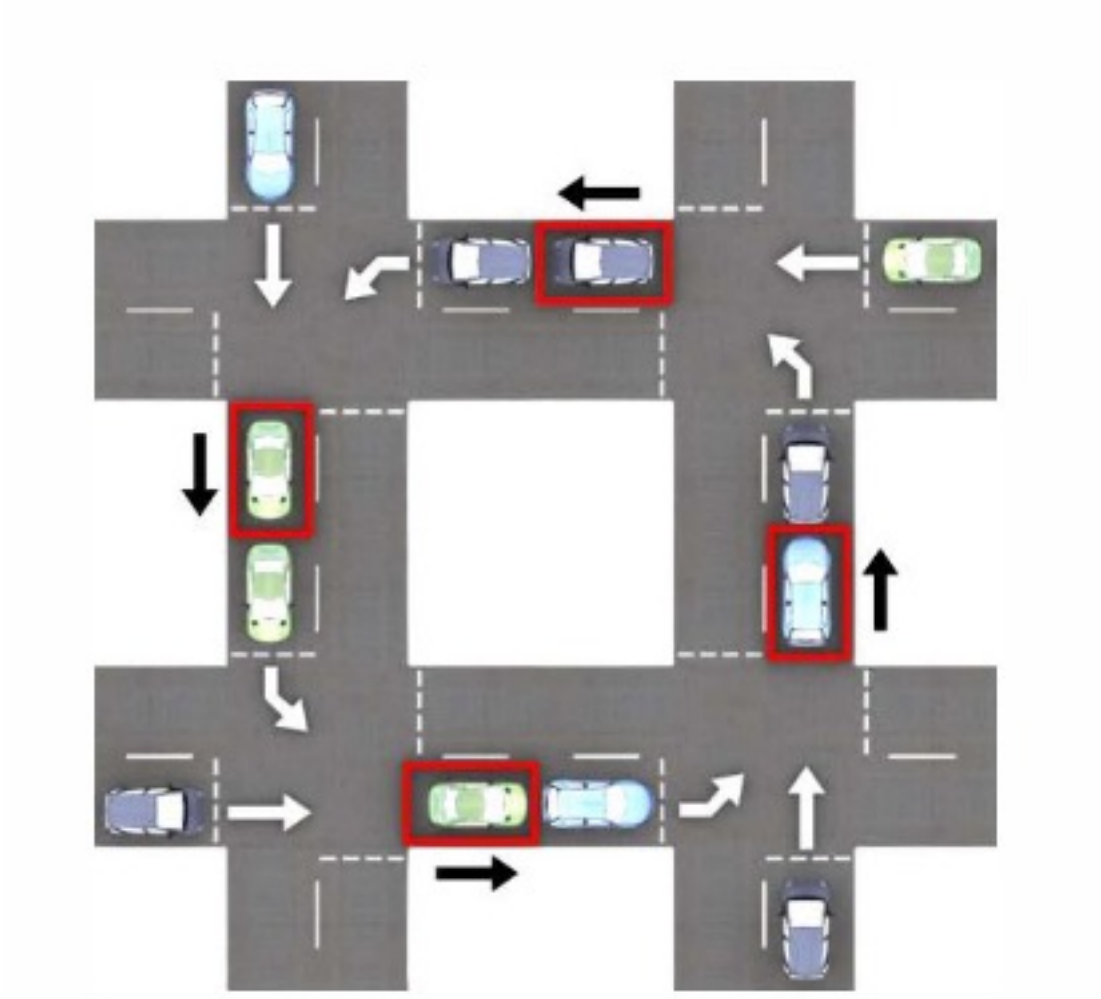
https://en.wikipedia.org/wiki/Wormhole_switching

Deadlock in routing

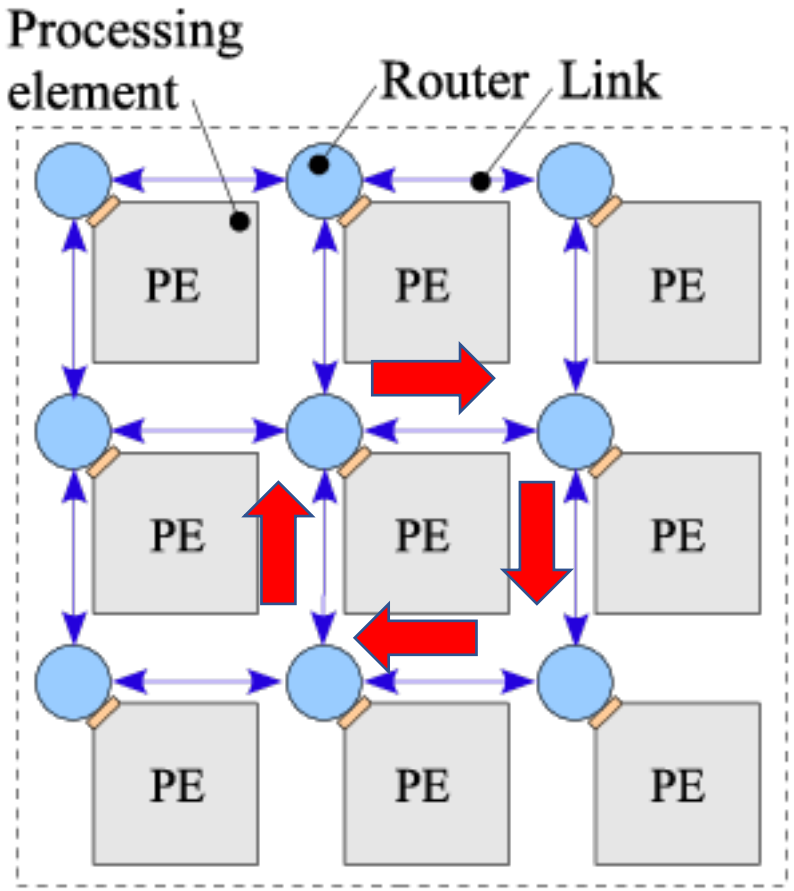


- Deadlock can happen on any NoC
  - Communication is stuck forever
- Deadlock-free routing necessary
  - Turn model based
  - Prevent one of the turns

Deadlock in real life

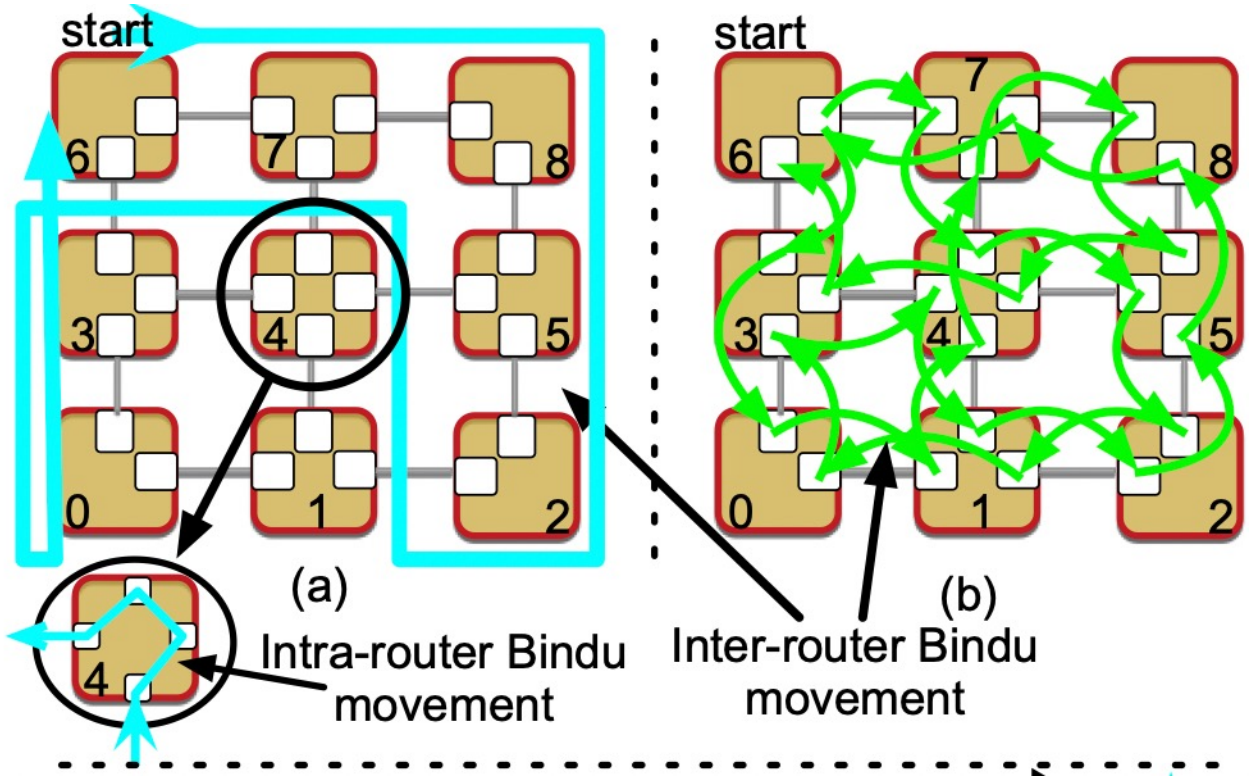


Solving Deadlock (1)



- Design routing algorithms that do not have deadlock
- Example: XY, YX routing

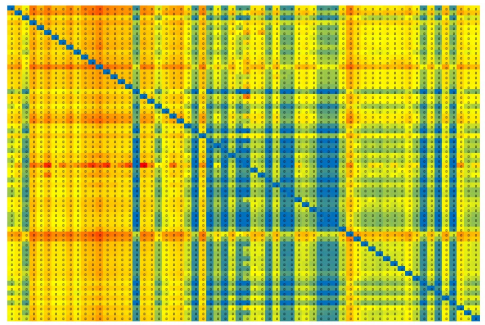
Solving Deadlock (2)

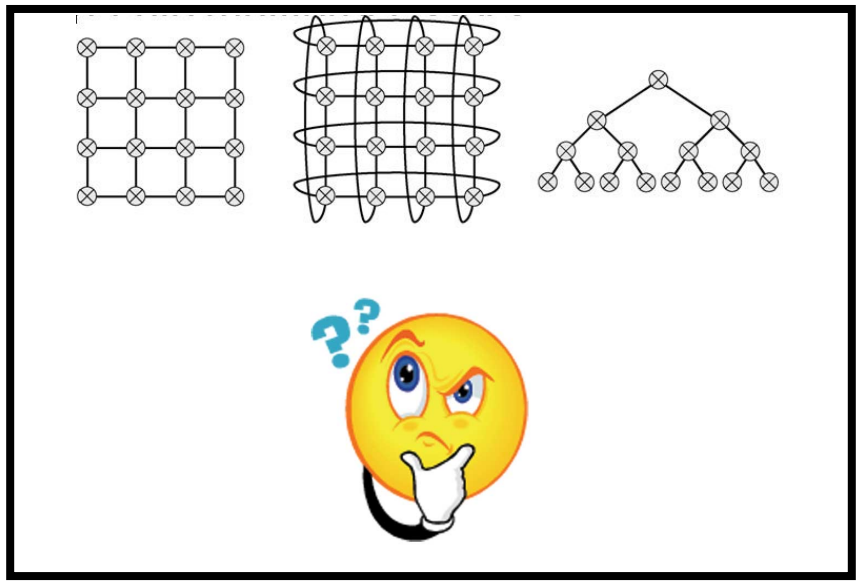


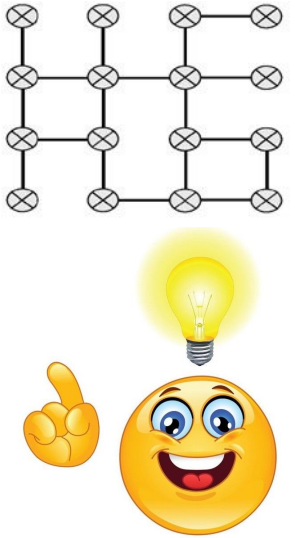
- **Bindu method**
- **Imagine a bubble moving**

Mayank Parasar and Tushar Krishna. 2019. BINDU: deadlock-freedom with one bubble in the network. In Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip (NOCS '19). Association for Computing Machinery, New York, NY, USA, Article 3, 1–8. DOI:https://doi.org/10.1145/3313231.3352359

Application-specific NoC design

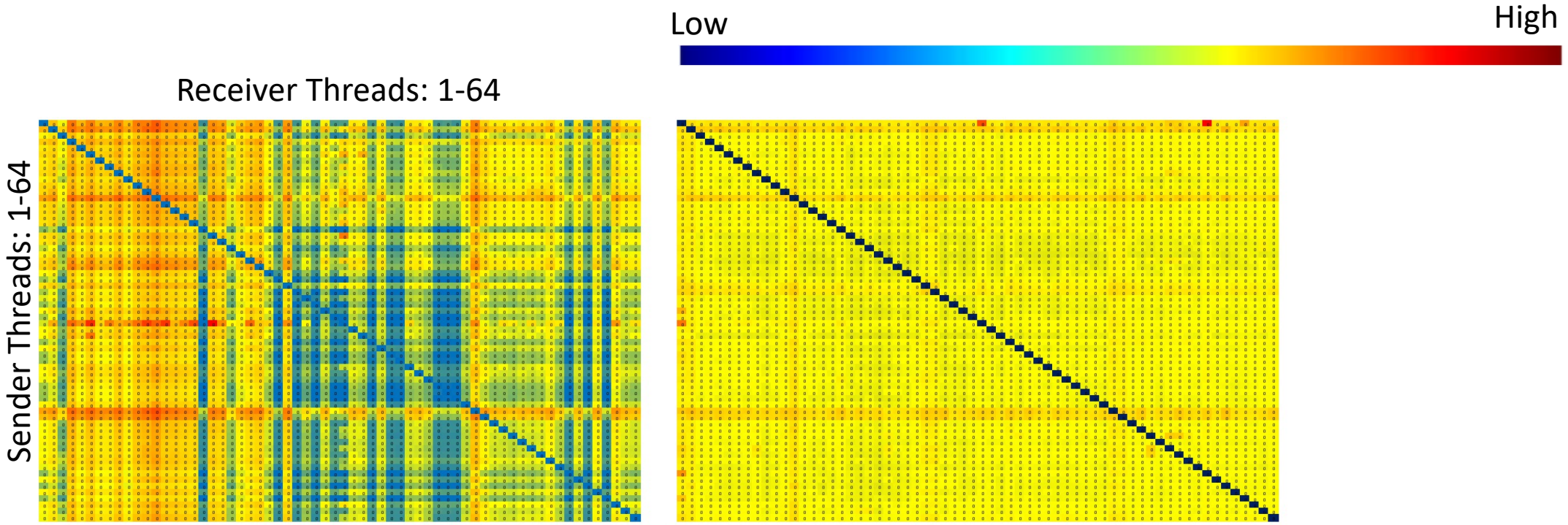






- Given an application behavior, what is the best NoC that we can design?
- Application behavior represented using traffic patterns here

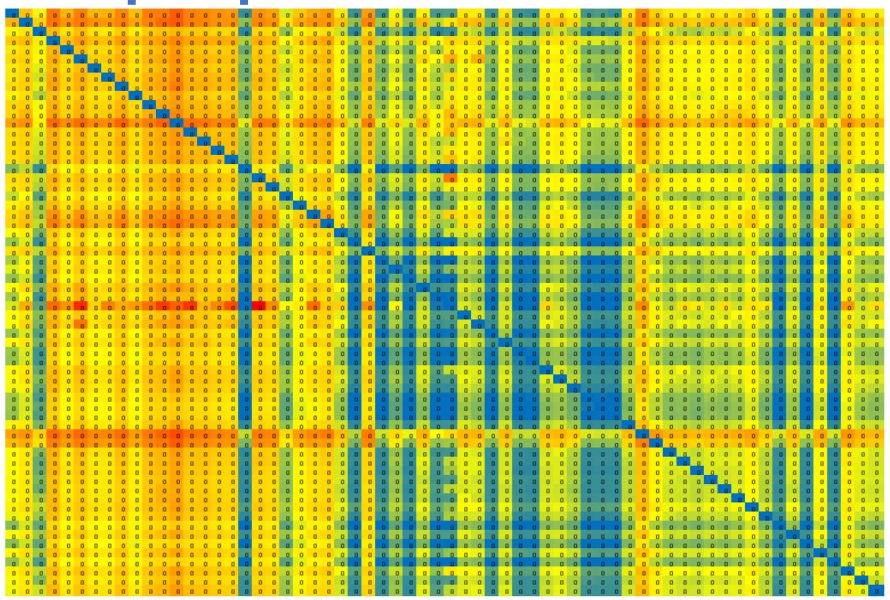
Traffic patterns

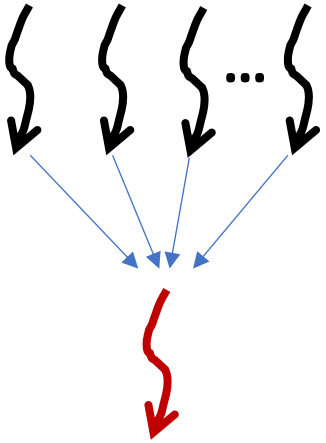


- **Different applications have different traffic patterns**
  - **Influences NoC design**
  - **Canneal: Few long-range links**
  - **FFT: Uniform traffic, mesh may be sufficient**

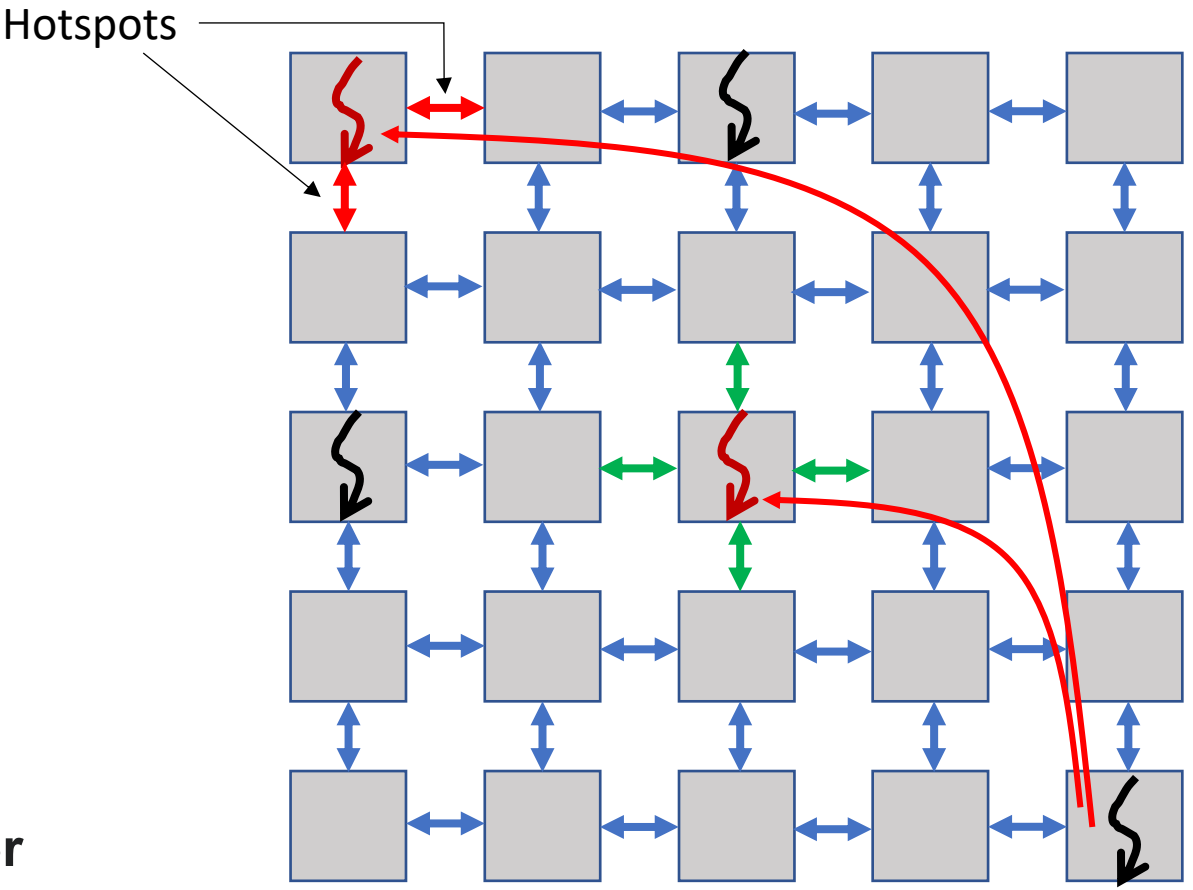
Traffic pattern aware NoC design

Receives data from every thread



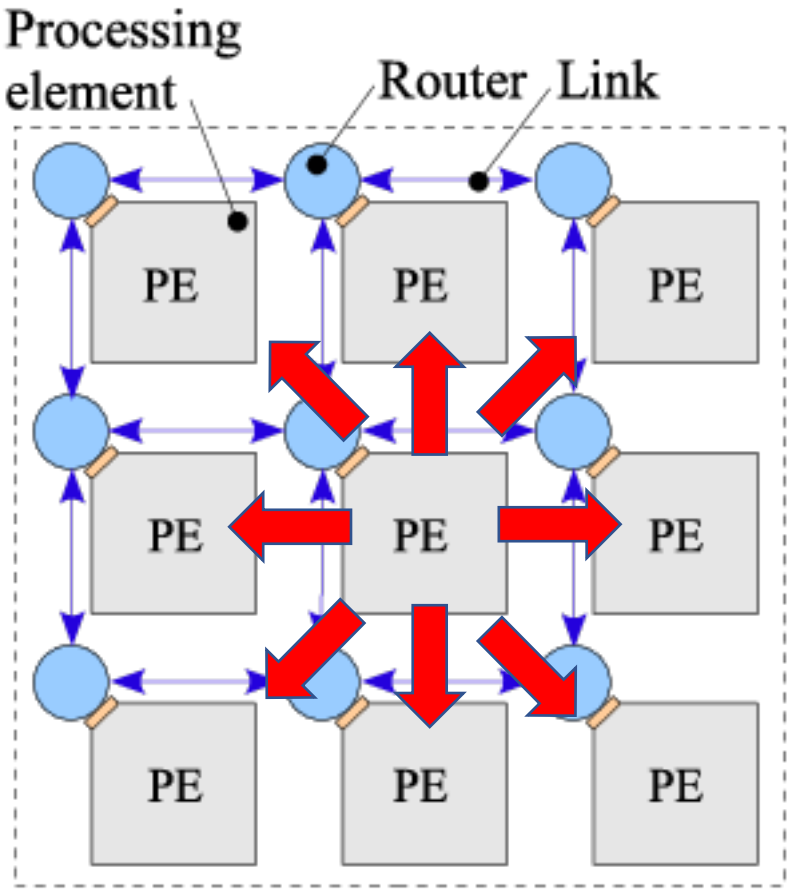


- Case-1: Map to corner
  - Only 2 links to reach there
  - Longest path: 8 hops
- Case-2: Map to center
  - 4 links to reach there
  - Longest hop: 4 hops



Longest path: 8 hops

Multicast and Broadcast



- Multicast
  - One-to-many traffic
- Broadcast
  - One-to-all traffic
- Observed in some applications and cache coherence protocols
- Also influences choice of NoC design

Many-to-few-to-many traffic

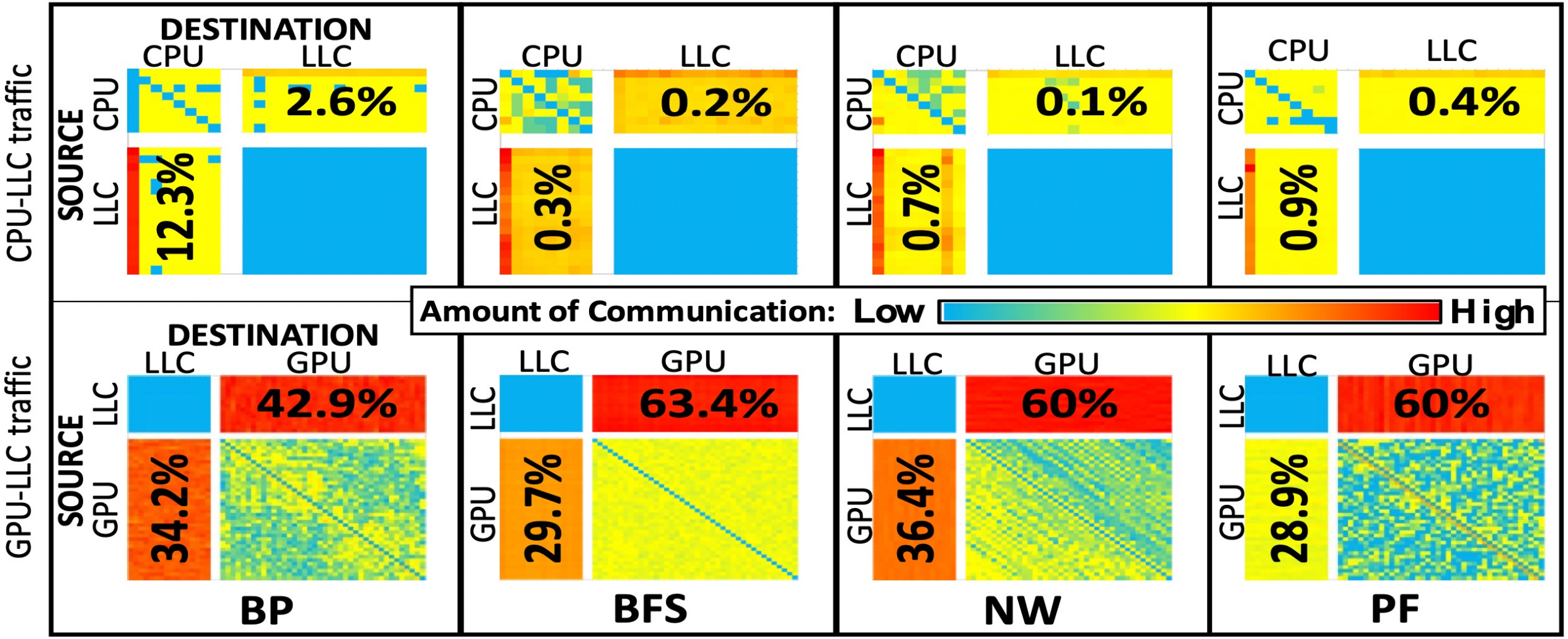


Fig. 1. Traffic pattern heat map for four applications (BP, BFS, NW, and PF) running on a 64-tile heterogeneous manycore system. The numbers indicate percentage of total traffic contributed by that section (*e.g.*, CPU-LLC communication results in 2.6% of total traffic for BP).

Application Behavior: ML

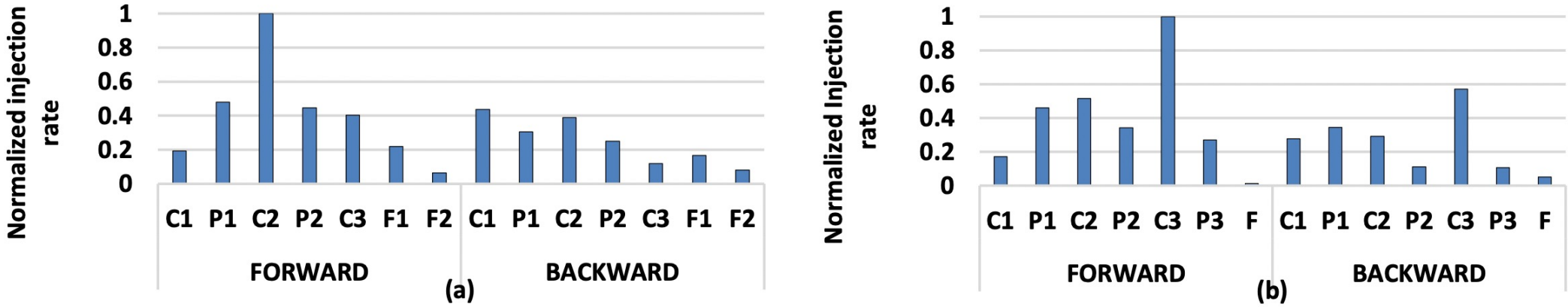


Fig. 5. Message injection rate for training the (a) LeNet and (b) CDBNet (normalized with respect to the highest injection rate layer).

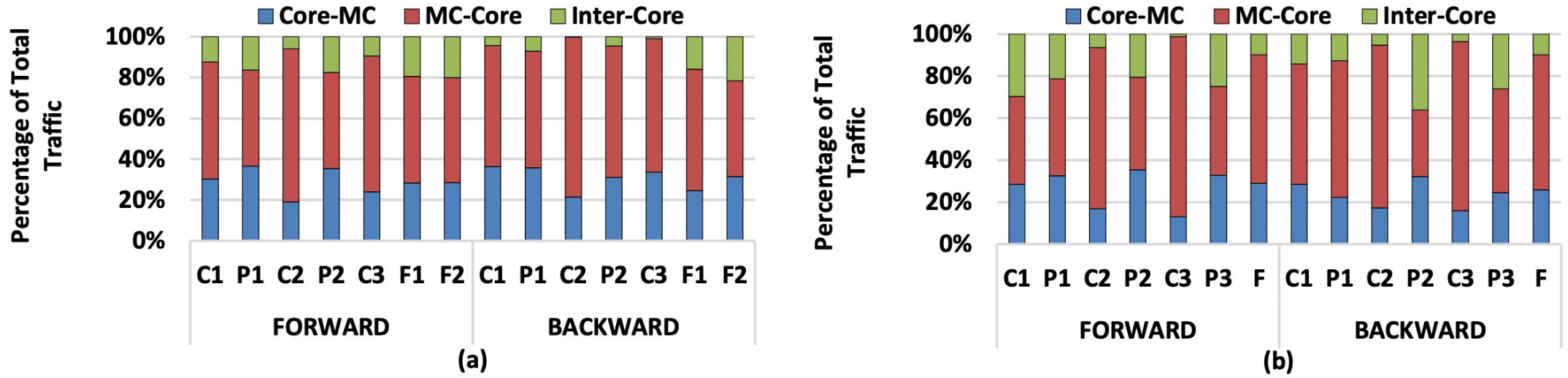
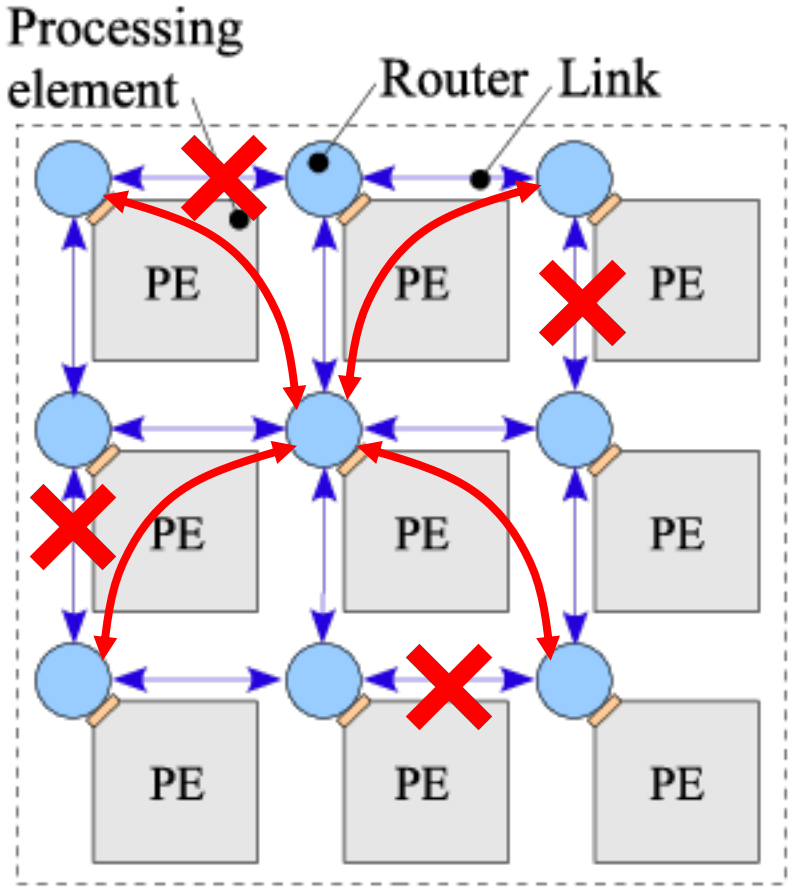
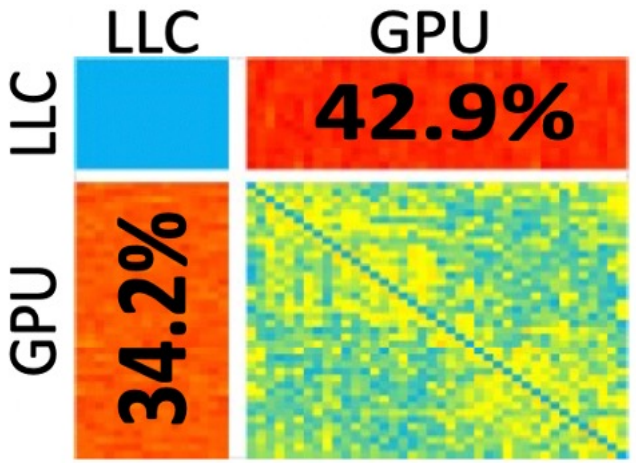


Fig. 6 Traffic breakdown for each layer of the (a) LeNet and (b) CDBNet CNNs. The graphs aggregate the CPU and GPU traffic (collectively referred to as "Core") to emphasize the heavy many-to-few traffic and the asymmetric nature of the traffic to/from the MCs for training CNNs.

NoC design for Many-to-few-to-many traffic





- Use links more judiciously
- Concentrate links in few locations
  - Good for many-to-few traffic
- Can be used for ML applications

Other Application Behaviors

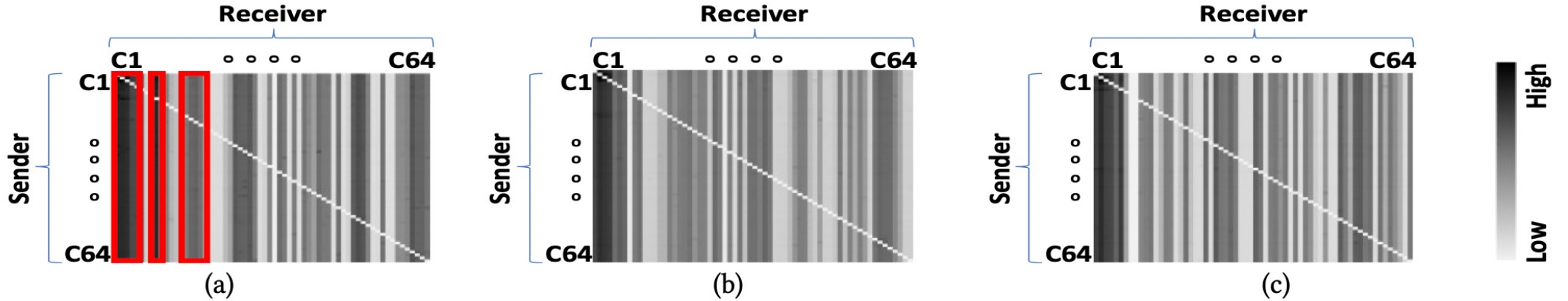
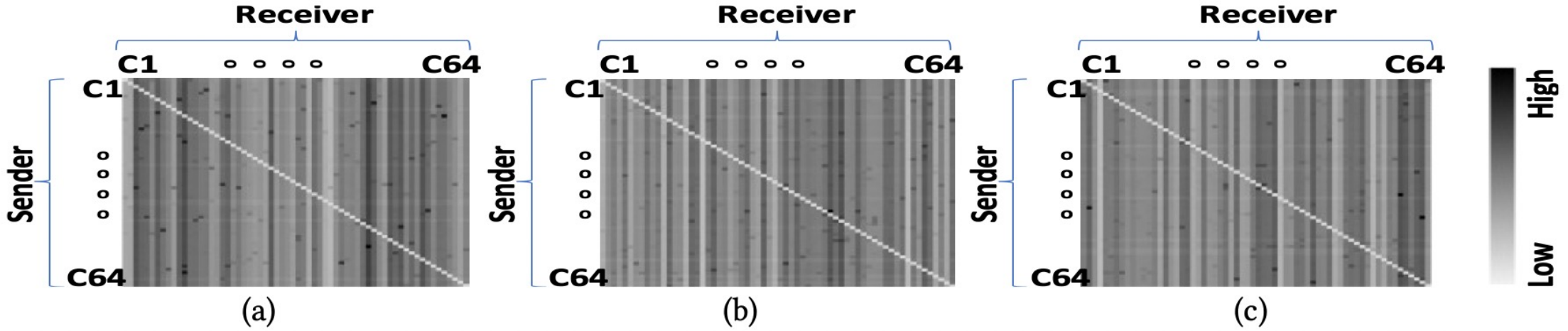
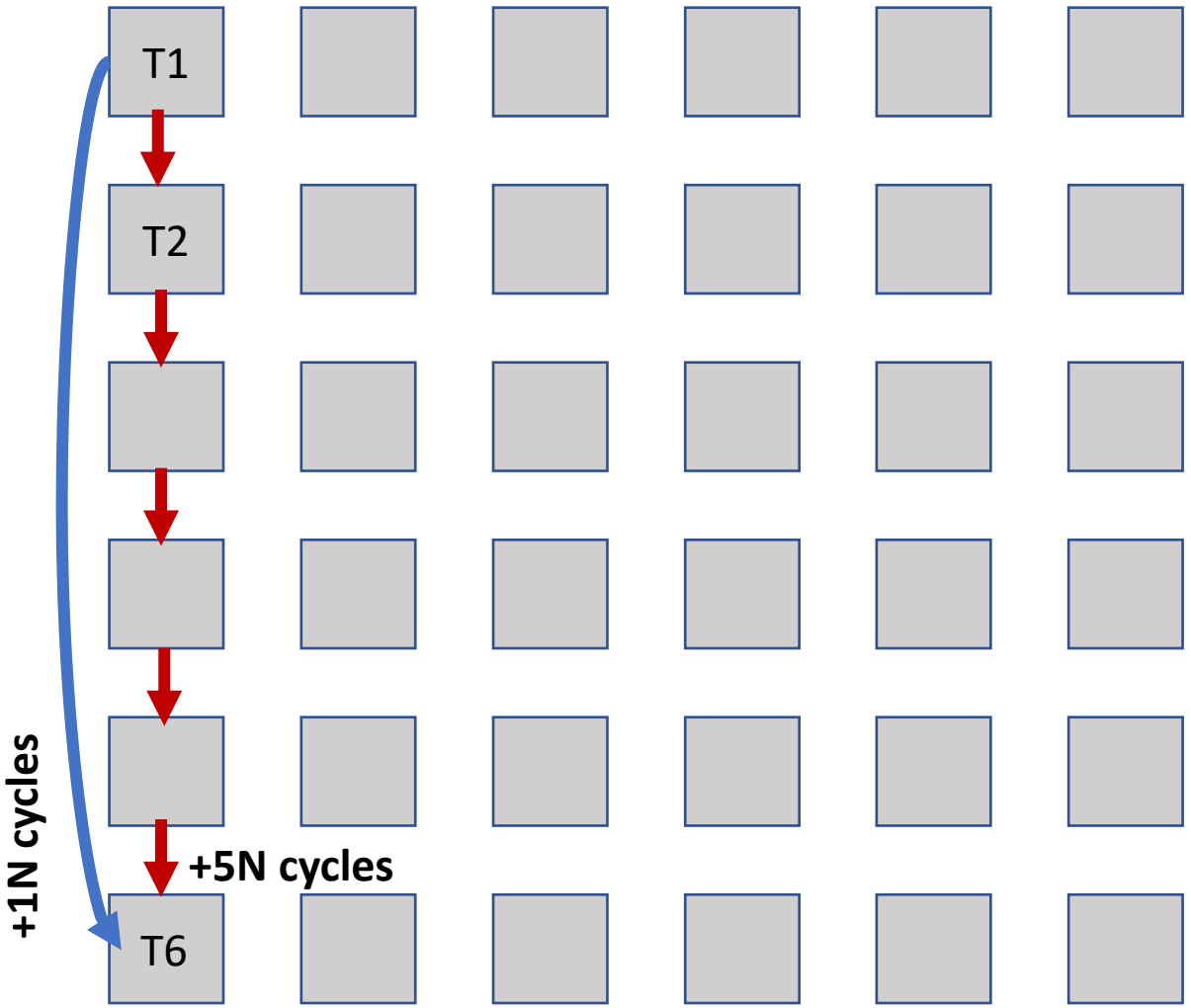


Fig 3: CPU-to-CPU communication profile for Gerbil in the form of heat map for input datasets (a) *E. Coli*, (b) *Prochlococcus sp.*, and (c) *Vibrio cholerae* (C_i: CPU*i*; Red boxes highlight *few* of the patches of heavy communication)

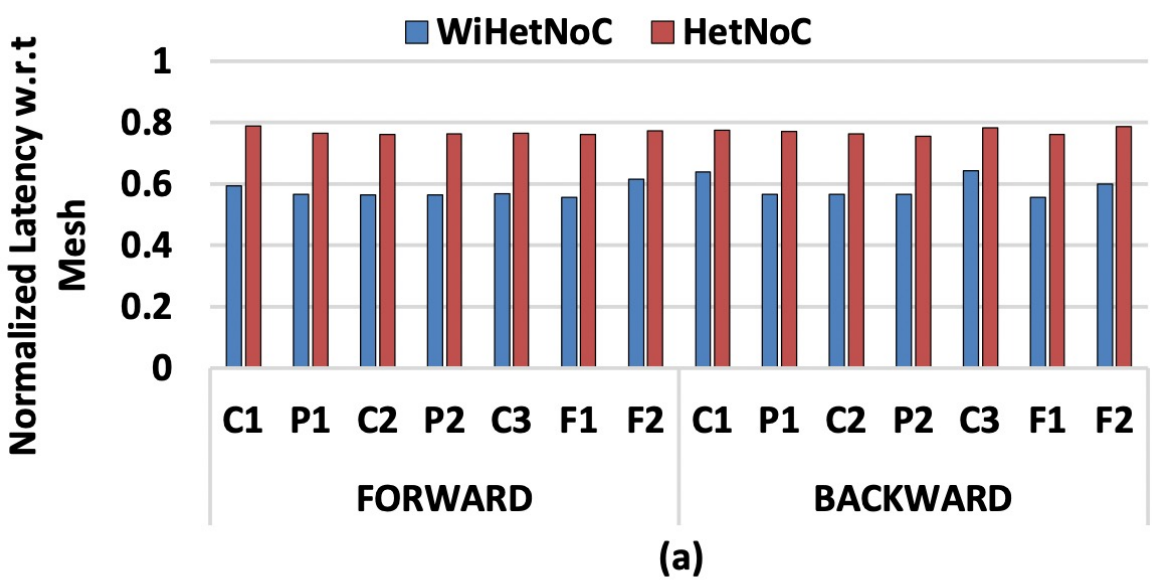


NoC Design in Manycore (1)



- Application behavior known
  - Case-1: T1 only communicates with T2
  - Case-2: T1 only communicates with T6
- Application-specific NoC design
  - Given N links
  - C(C(N,2),L) possible ways
  - How to place these links?

Why design better NoCs: Performance



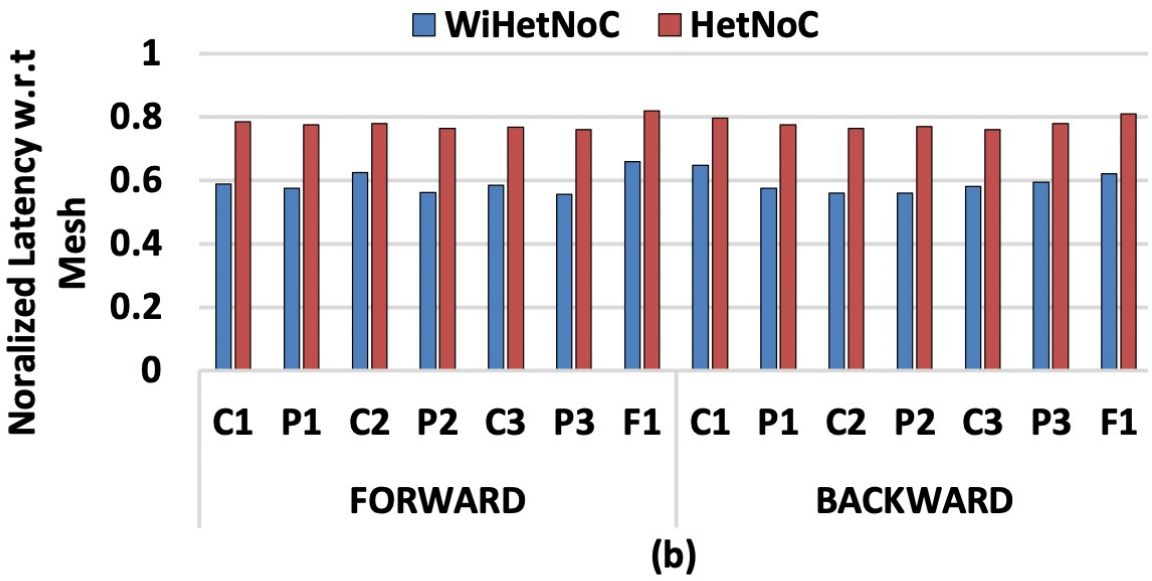


Fig. 17. Normalized network latency for training the (a) LeNet and (b) CDBNet.

- Bad NoC design can bottleneck performance
- Good NoC leads to better performance

Why design better NoCs: Temperature

- Congestion can lead to higher power & higher temperature
- Divert some of the traffic to other places
- Can reduce temperature

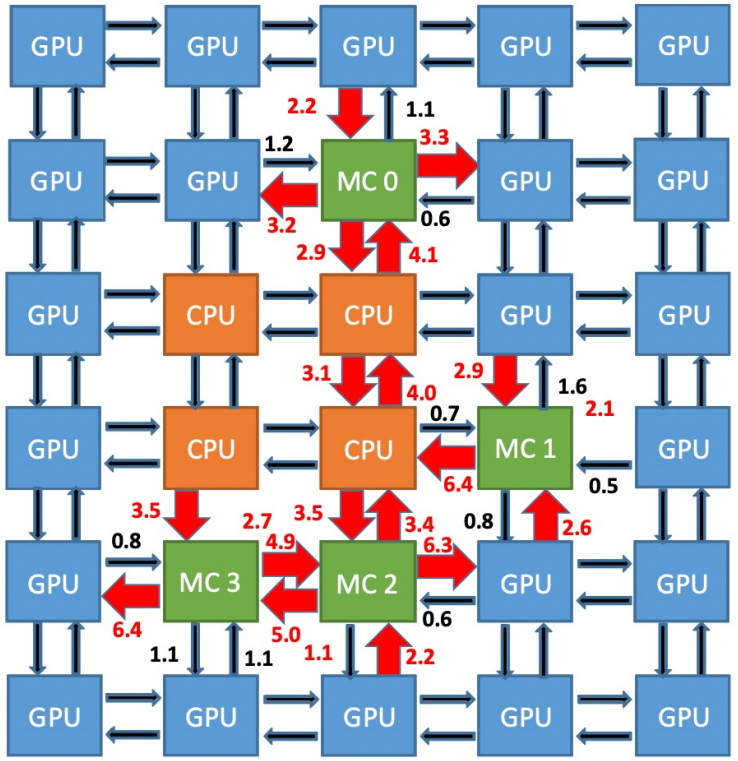
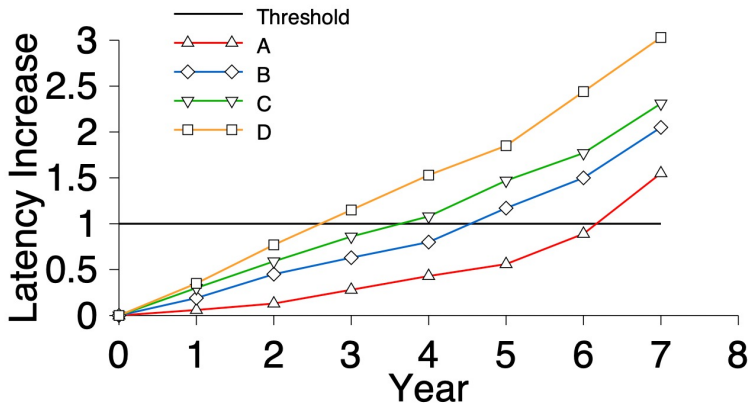
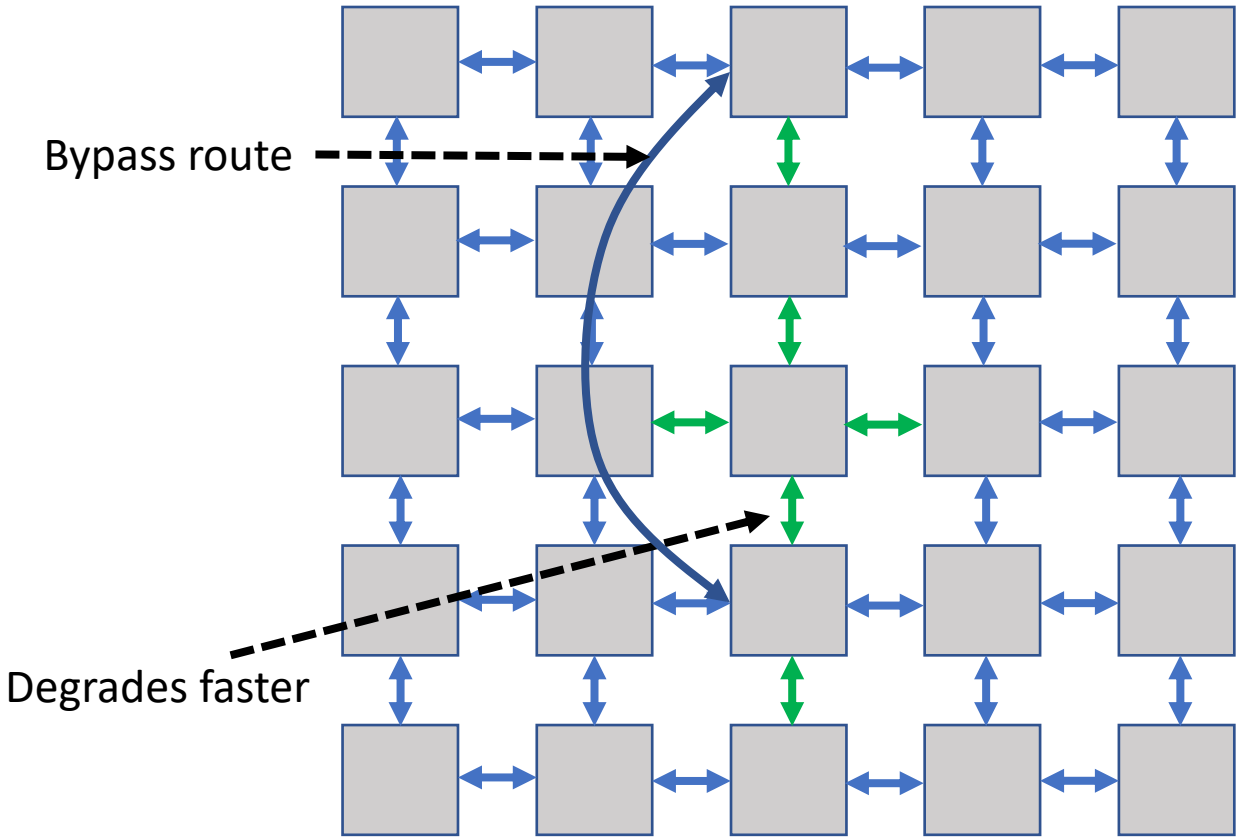


Fig. 8. The central 6×5 portion of the optimized Mesh NoC indicating the link utilizations and the locations of CPUs and MCs for LeNet computations. All link utilizations are normalized with respect to the mean link utilization. The red arrows indicate the bandwidth bottlenecks whose utilization is at least 100% more than the mean.

W. Choi et al., "On-Chip Communication Network for Efficient Training of Deep Convolutional Networks on Heterogeneous Manycore Systems," in IEEE Transactions on Computers, vol. 67, no. 5, pp. 672-686, 1 May 2018, doi: 10.1109/TC.2017.2777863.

Why design better NoCs: Aging

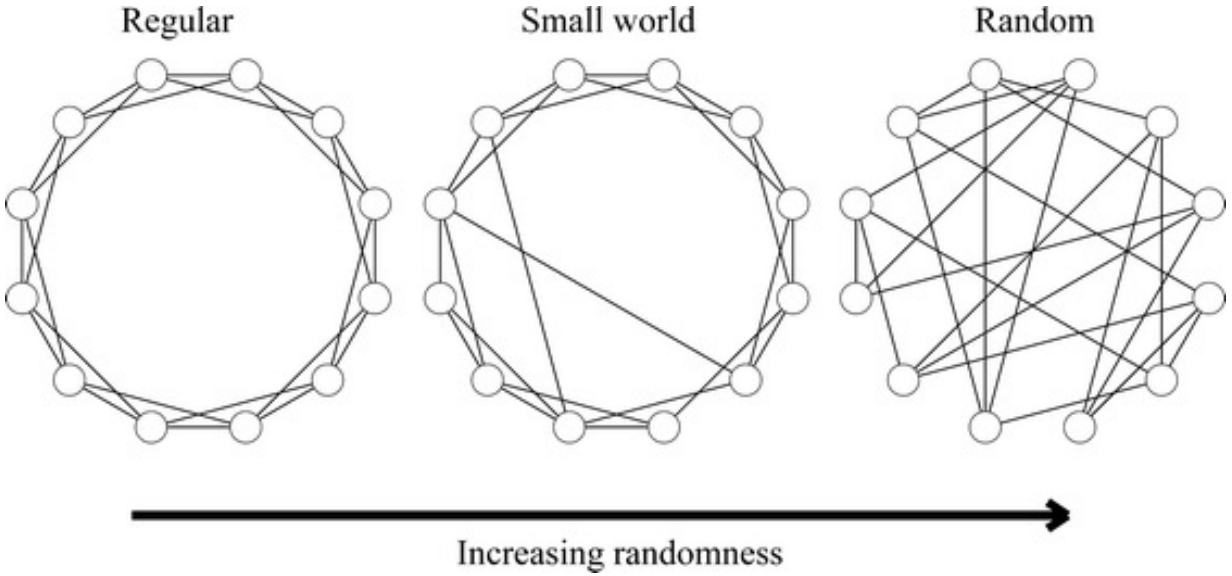


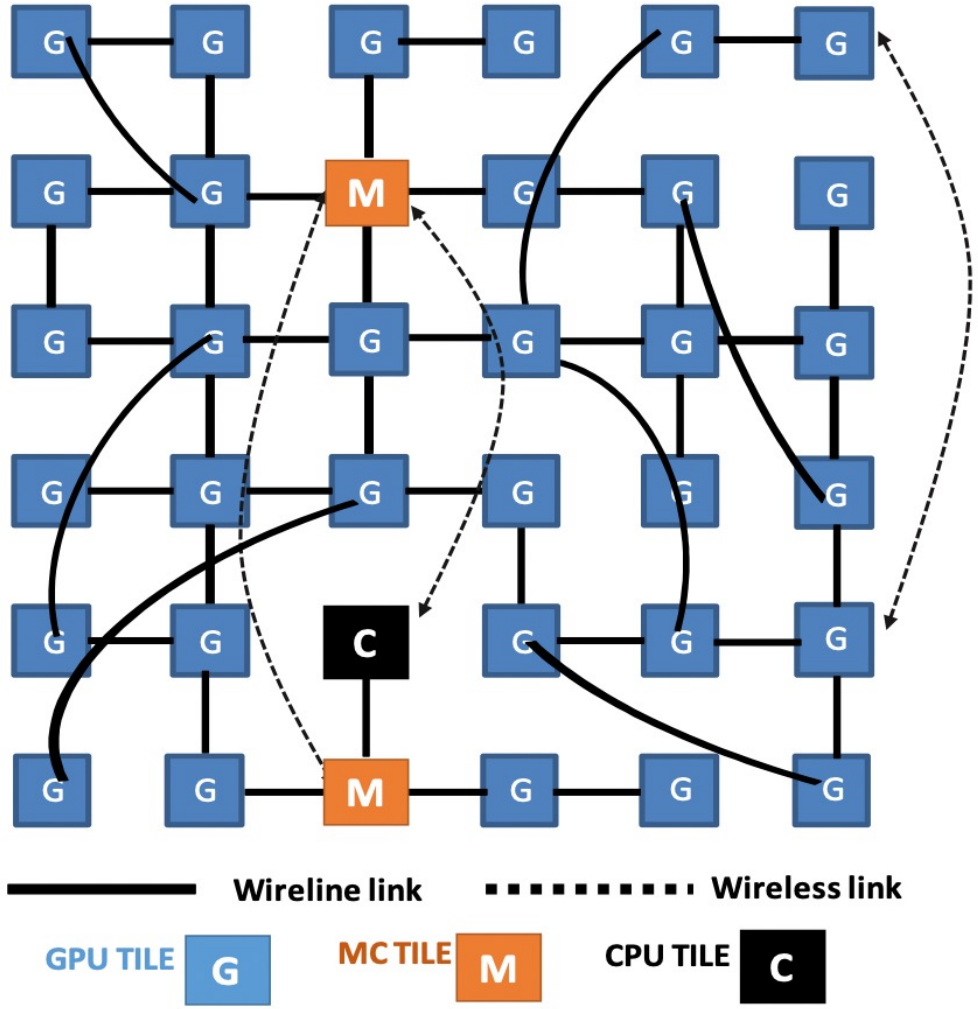


- Overutilization accelerates ageing
- Links get slower as a result
- Bypass links can ease the burden on overutilized links

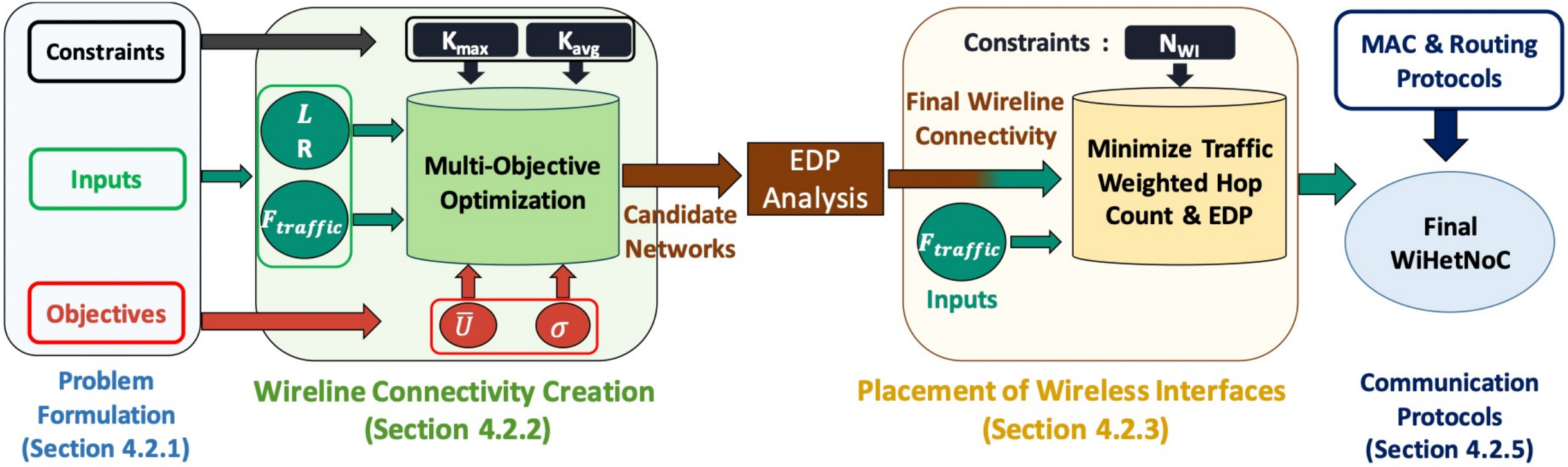
Irregular NoC design

- Small-world NoC works better than regular ones
- Few application specific shortcuts





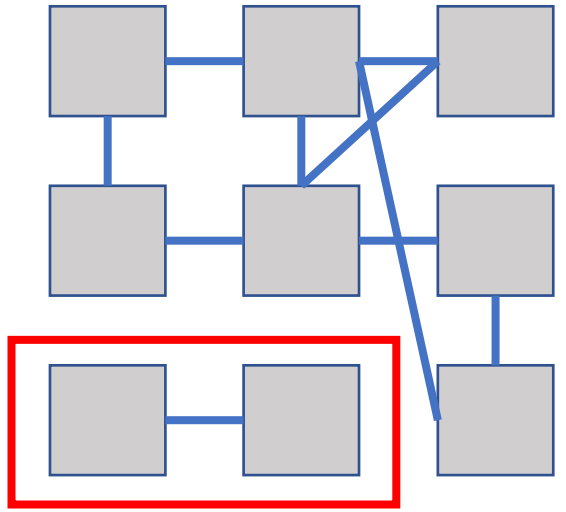
Overall process



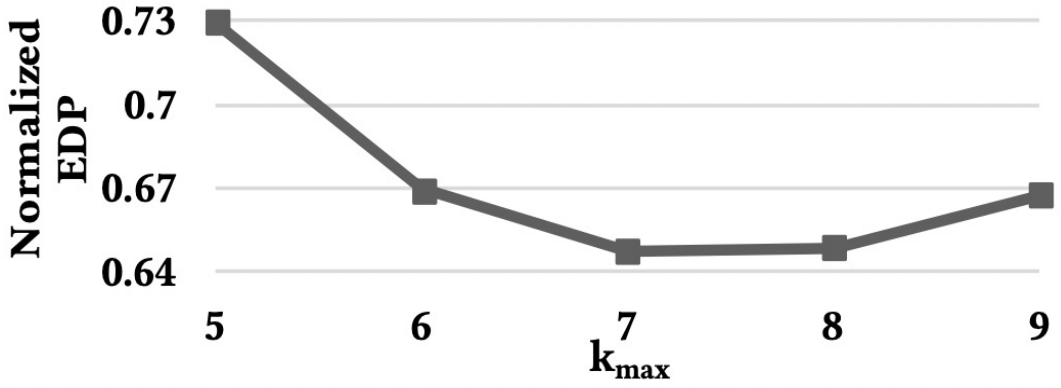
K_{max} = Maximum port count for a router; L = Number of NoC links; $F_{traffic}$ = Traffic pattern; N_{WI}= Number of WIs
$\bar{U}$ = Link utilization average, σ = Link utilization Standard deviation

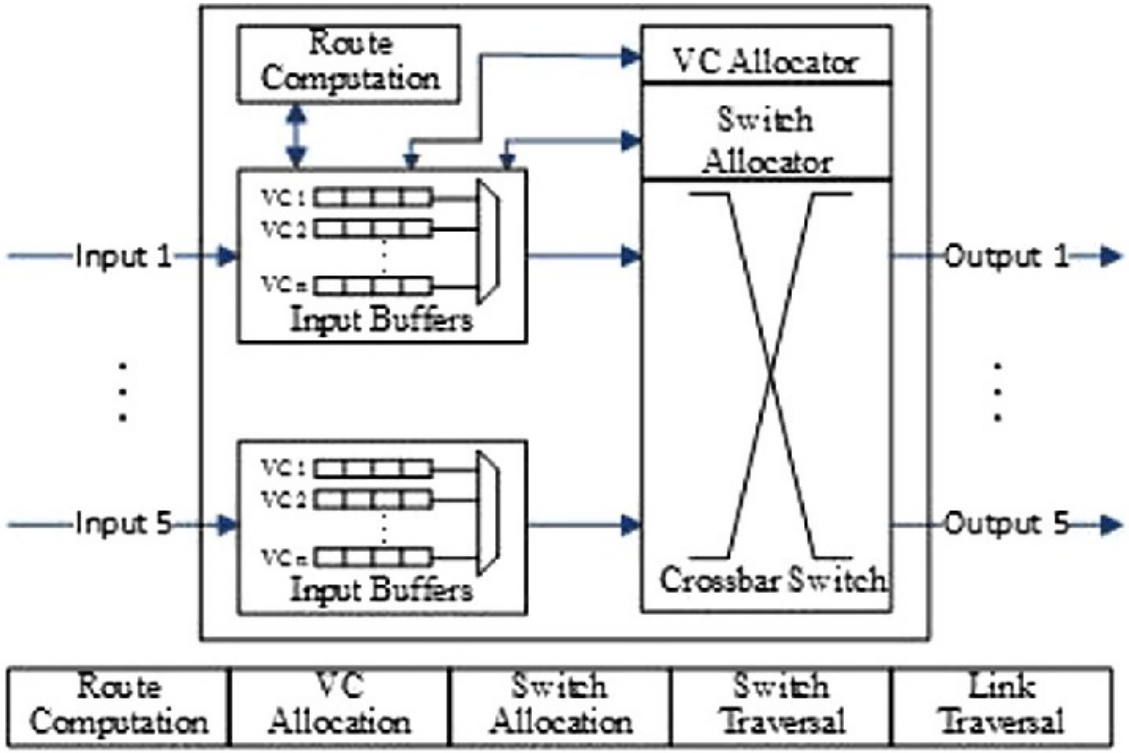
Rules of NoC design(1)

- All cores must be connected
- Paths must exist to communicate between any pair of cores
- Islands of cores not permitted



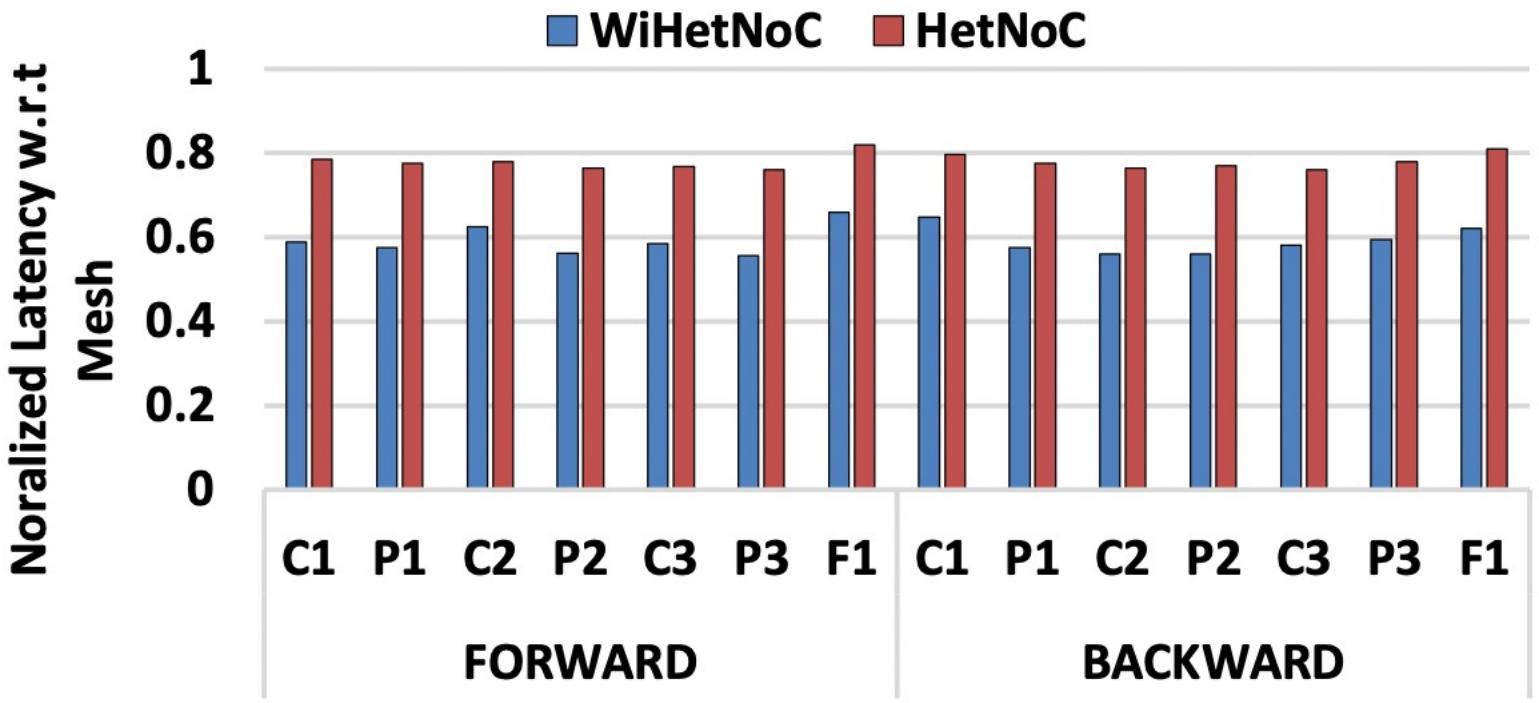
Rules of NoC design(1)





- Routers cannot be too large
- Large routers are counter-productive
- More energy, complex design

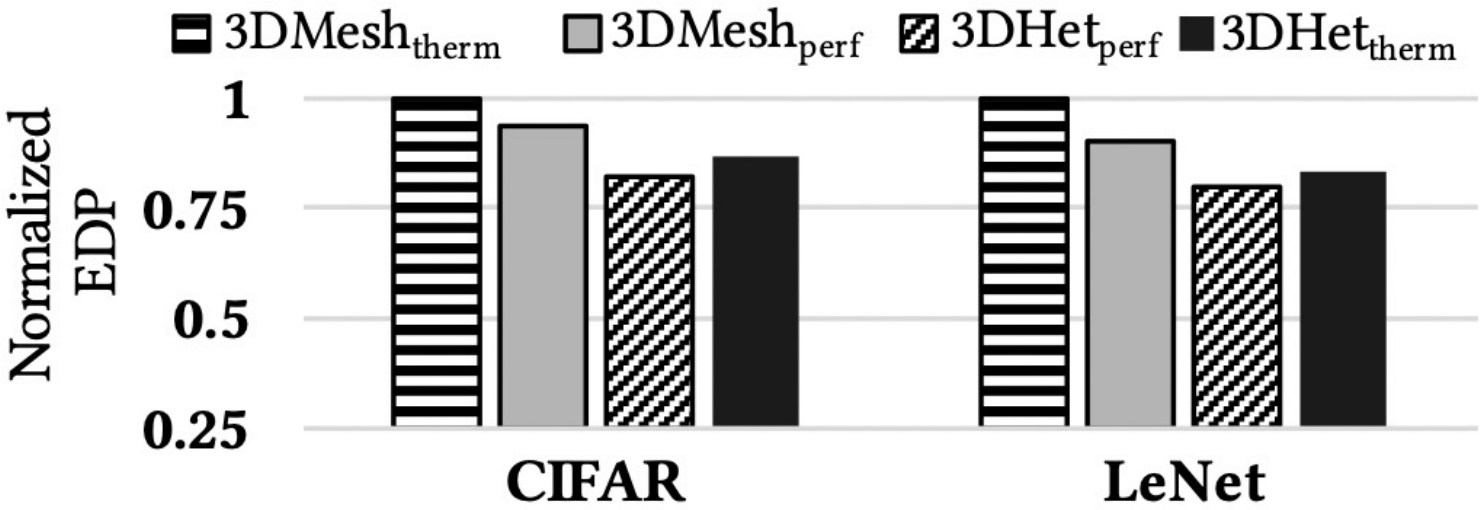
NoC results (1)



- Optimized NoC leads to better performance
- lower execution time for ML than mesh NoC

W. Choi et al., "On-Chip Communication Network for Efficient Training of Deep Convolutional Networks on Heterogeneous Manycore Systems," in IEEE Transactions on Computers, vol. 67, no. 5, pp. 672-686, 1 May 2018, doi: 10.1109/TC.2017.2777863.

NoC results (2)



- **3D NoCs are even better**
- **Better performance, (both execution time and energy)**

B. K. Joardar et al., "3D NoC-enabled heterogeneous manycore architectures for accelerating CNN training: Performance and thermal trade-offs," 2017 Eleventh IEEE/ACM International Symposium on Networks-on-Chip (NOCS), 2017, pp. 1-8.