Asymmetry in Butterfly Fat Tree FPGA NoC

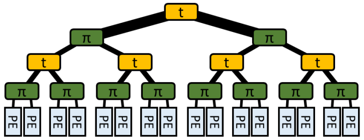
Dongjoon(DJ) Park, Zhijing Yao, Yuanlong Xiao, André DeHon

Implementation of Computation Group
University of Pennsylvania

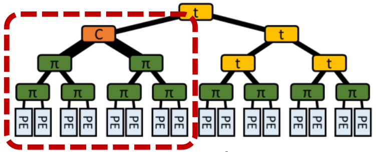
Penn Engineering
University of Pennsylvania

Story

- Problem
  - Traditional Butterfly Fat Tree (BFT) NoC is symmetric; when graph workloads are unbalanced, we can't selectively allocate more bandwidth
- Idea
  - **Asymmetric BFT**
  - Allocates more bandwidth to specific nodes yet uses *similar resources* with symmetric BFT
- What Asymmetric BFT will deliver
  - Expands the design space of Soft NoCs; users can tailor the NoC to their applications, fully exploiting FPGA's reconfigurability
- Result
  - Up to 32% improvement in throughput on realistic workloads



<Symmetric BFT>



<Asymmetric BFT>

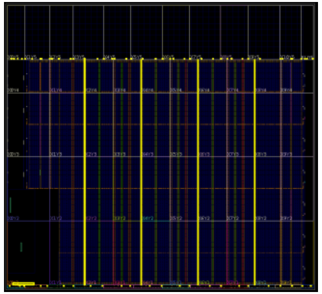
Table of Contents

- Background
- Motivation
- Idea
- Evaluation
- Conclusion

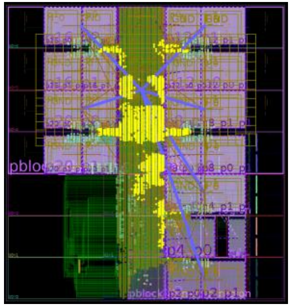
Table of Contents

Background

- **Network-on-Chip (NoC)** on FPGA
  - **Hard NoC**: embedded NoC on the FPGA (e.g. AMD Xilinx Versal, Achronix Speedster7t)
    - Better performance
    - Compact, don't use programmable resources
  - **Soft NoC**: an overlay NoC built on top of the commercial FPGA
    - More flexibility



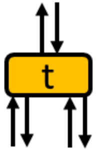
<Hard NoC example, AMD Versal>

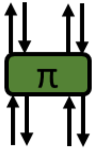


<Soft NoC example>

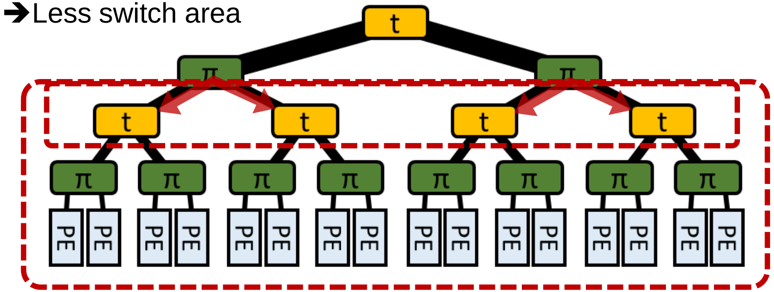
Background

- **Butterfly Fat Tree (BFT)** soft NoC
  - Bandwidth of each level of BFT can be configured by properly selecting *t switches* and *pi switches*





Less bandwidth
➔Less switch area



More bandwidth
➔ More switch area

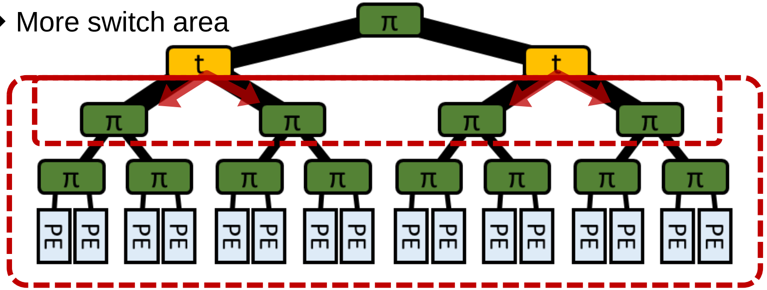
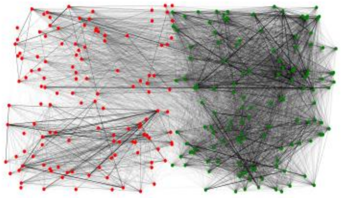


Table of Contents

- Background
- Motivation
- Idea
- Evaluation
- Conclusion

Motivation

- Problem 1: Realistic workloads are sometimes unbalanced
- Problem 2: Traditional BFT is symmetric
  - Each level is homogeneously composed of either t switches or pi switches
  - When you want to allocate more bandwidth to specific PEs, it requires more resources

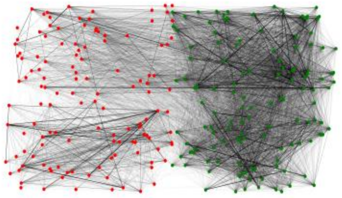


<Example of unbalanced workloads[1] after bi-partitioning>

[1] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” http://snap.stanford.edu/data, 2014

Motivation

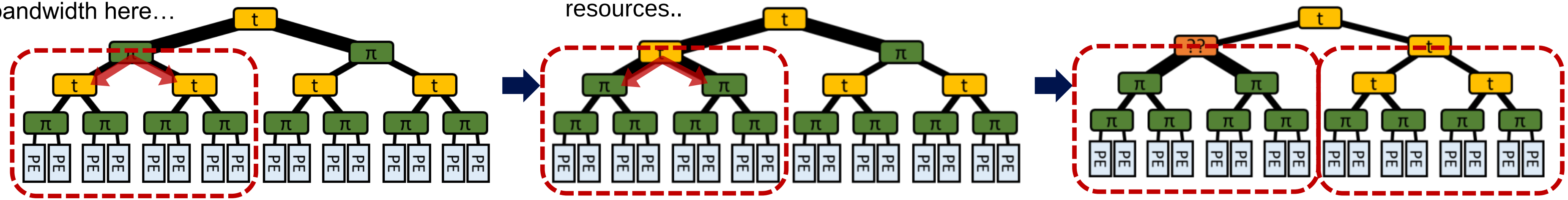
- Problem 1: Realistic workloads are sometimes unbalanced
- Problem 2: Traditional BFT is symmetric
  - Each level is homogeneously composed of either t switches or pi switches
  - When you want to allocate more bandwidth to specific PEs, it requires more resources



<Example of unbalanced workloads[1] after bi-partitioning>

If we want to allocate more bandwidth here…

Using more resources..



num t switches: 4
num pi switches: **6**
*Note: multiple switches are overlapped in the fig above

num t switches: 4
num pi switches: **8**

As we use more resources here

Compensate here

- What we want: selectively allocate more bandwidth to some PEs using similar resources

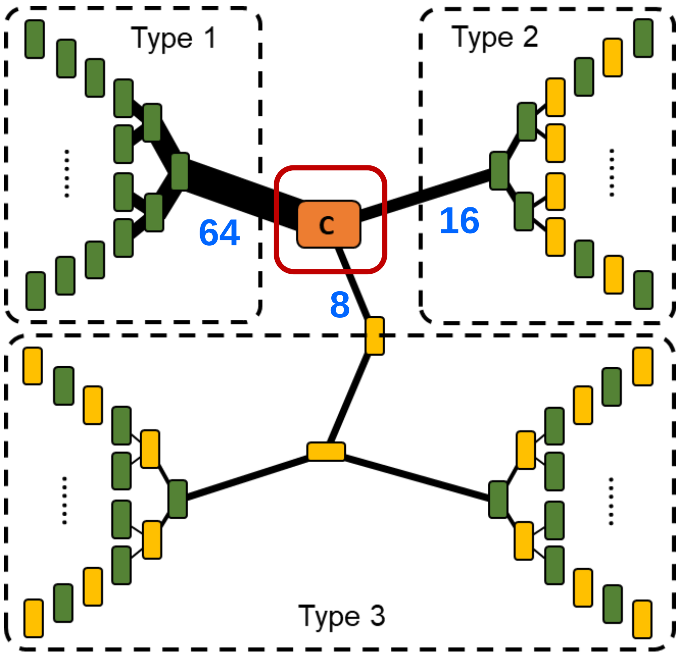
[1] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, 2014

Penn Engineering
University of Pennsylvania

Table of Contents

Idea

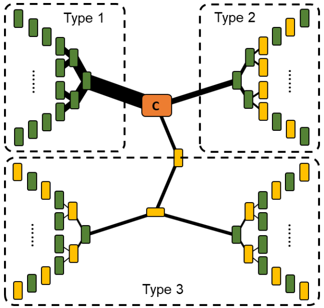
- Asymmetric BFT: different bandwidth with similar resource usage
  - Example Asym BFT
    - Type 1: most dense
    - Type 2: dense
    - Type 3: sparse
  - Converging switch

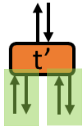


<Asymmetric BFT example>

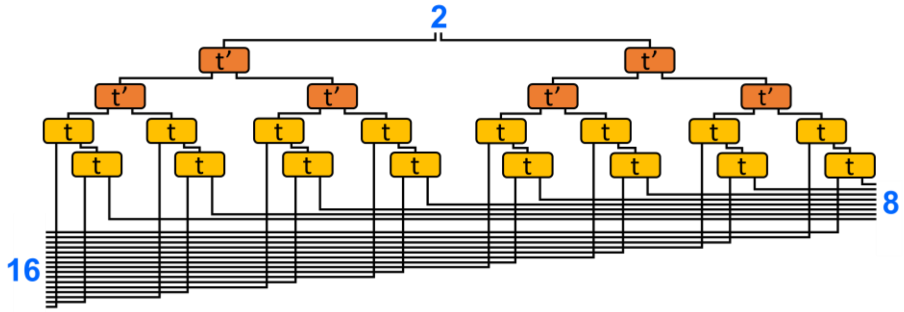
Idea

- Converging switch
  - Matches different bandwidths
  - Build with t switches only? ➔ Traffic congestion
  - Build with t switches and ***t-random* switches**
    - When downward, packet is directed lower left one cycle, and lower right another cycle





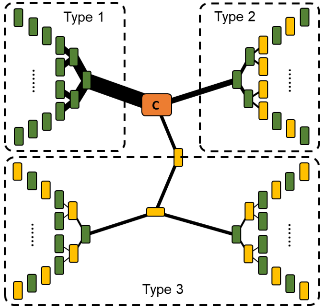
<t-random switch>

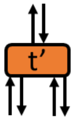


<Converging switch built with t switches and t-random switches>

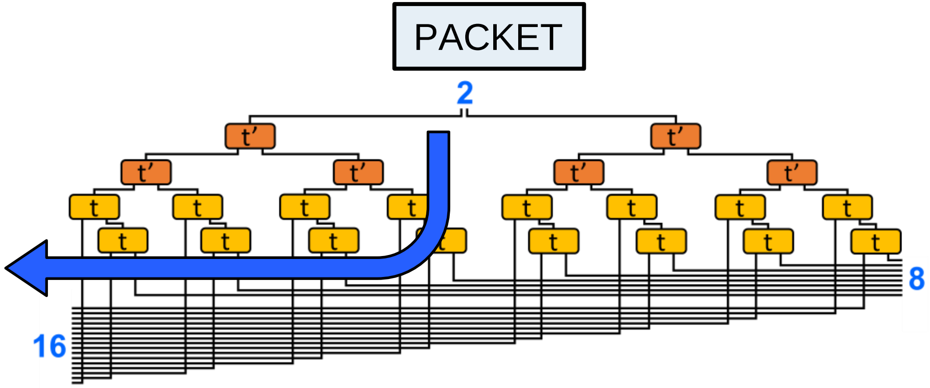
Idea

- Converging switch
  - Matches different bandwidths
  - Build with t switches only? ➔ Traffic congestion
  - Build with t switches and ***t-random* switches**
    - When downward, packet is directed lower left one cycle, and lower right another cycle





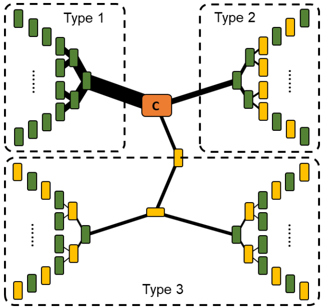
<t-random switch>

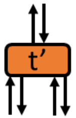


<Converging switch built with t switches and t-random switches>

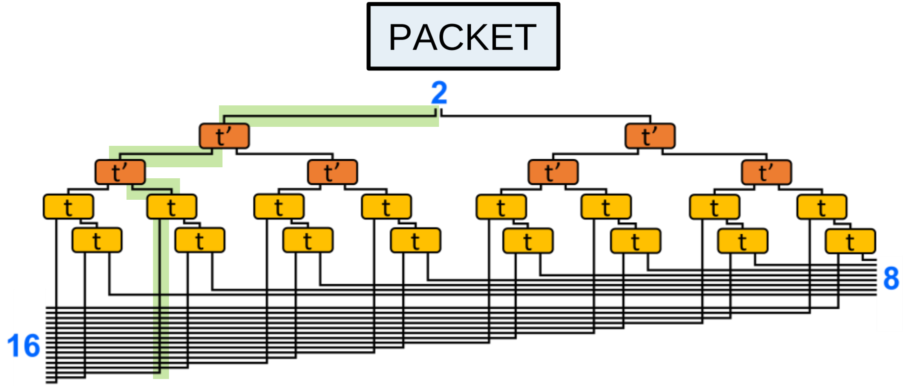
Idea

- Converging switch
  - Matches different bandwidths
  - Build with t switches only? ➔ Traffic congestion
  - Build with t switches and ***t-random* switches**
    - When downward, packet is directed lower left one cycle, and lower right another cycle





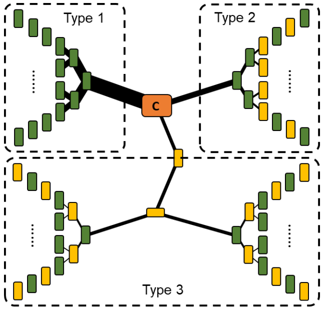
<t-random switch>

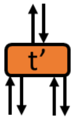


<Converging switch built with t switches and t-random switches>

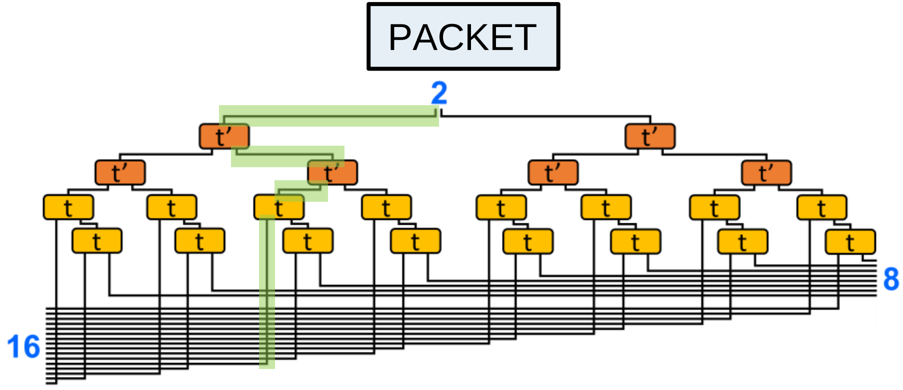
Idea

- Converging switch
  - Matches different bandwidths
  - Build with t switches only? ➔ Traffic congestion
  - Build with t switches and ***t-random* switches**
    - When downward, packet is directed lower left one cycle, and lower right another cycle





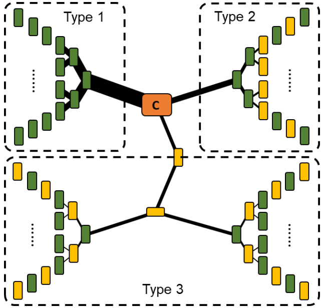
<t-random switch>

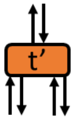


<Converging switch built with t switches and t-random switches>

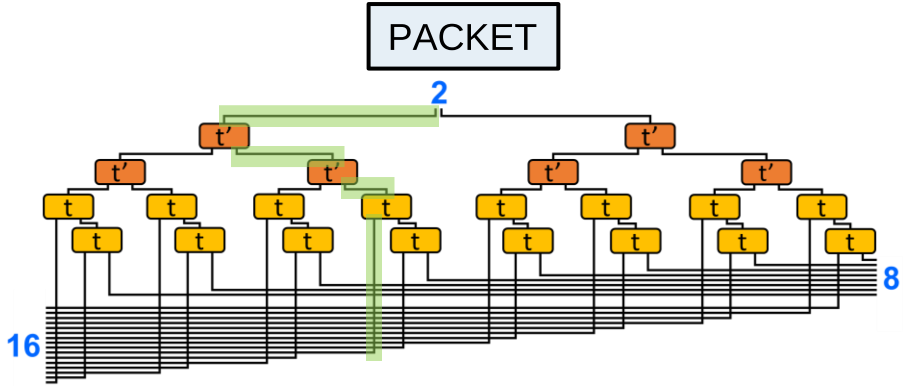
Idea

- Converging switch
  - Matches different bandwidths
  - Build with t switches only? ➔ Traffic congestion
  - Build with t switches and ***t-random* switches**
    - When downward, packet is directed lower left one cycle, and lower right another cycle





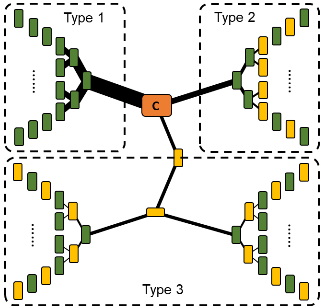
<t-random switch>

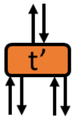


<Converging switch built with t switches and t-random switches>

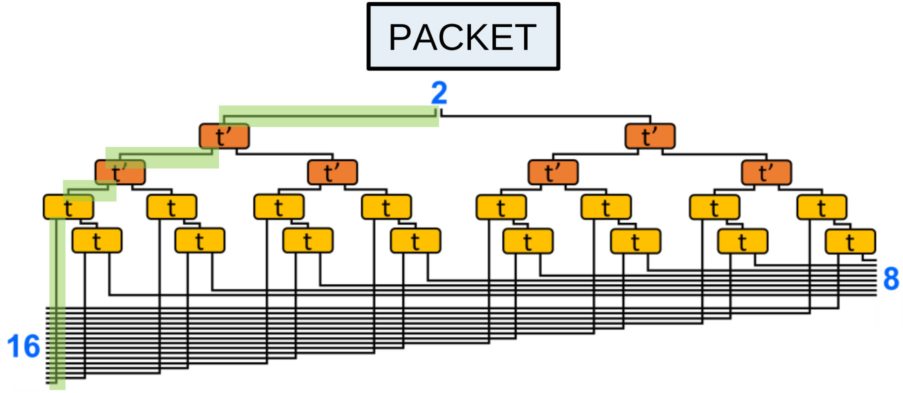
Idea

- Converging switch
  - Matches different bandwidths
  - Build with t switches only? ➔ Traffic congestion
  - Build with t switches and ***t-random* switches**
    - When downward, packet is directed lower left one cycle, and lower right another cycle





<t-random switch>



<Converging switch built with t switches and t-random switches>

Table of Contents

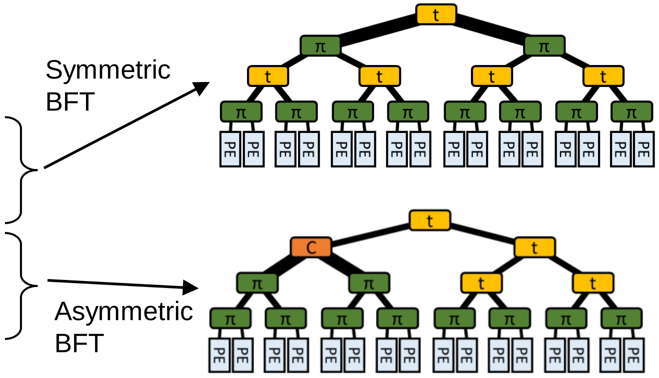
- Background
- Motivation
- Idea
- Evaluation
- Conclusion

Evaluation

- Simulation with iverilog
- Symmetric BFT vs Asymmetric BFT
  - Realistic graph workloads
  - Synthetic traffic patterns (omitted in the presentation)

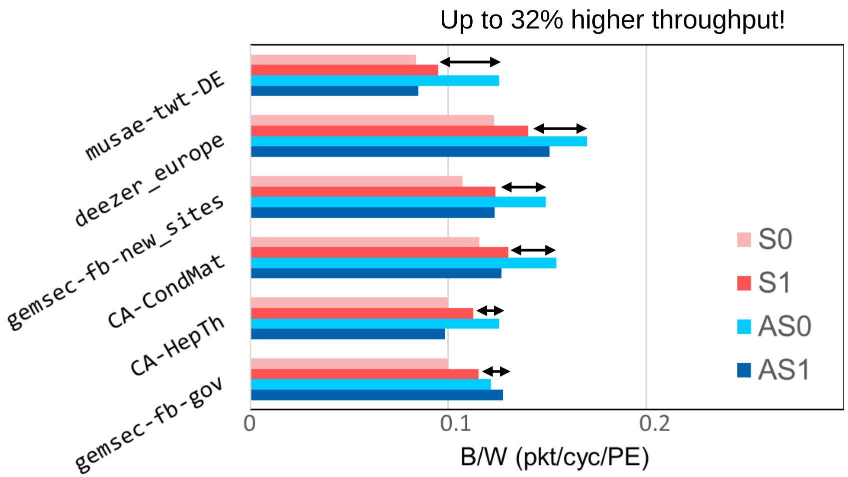
BFT-256	LUTs	Asymmetry
S0	122778	-
S1	143870	-
AS0	142896	Dense, dense, sparse, sparse
AS1	143029	Most dense, normal, sparse, sparse

Resource usage: AMD Vivado 2022.2



Evaluation

- Realistic graph workloads[1]
  - For balanced workloads: symmetric BFTs are better
  - For unbalanced workloads: asymmetric BFTs are better



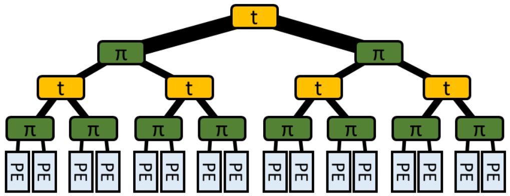
[1] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, 2014

Table of Contents

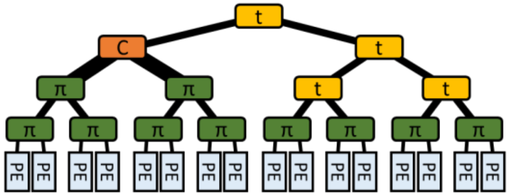
- Background
- Motivation
- Idea
- Evaluation
- Conclusion

Conclusion

- When the traffic is unbalanced, asymmetric BFT achieves up to 32%(realistic) and 76%(synthetic) more throughput than traditional symmetric BFT
  - Provides more options to the users with the similar resource usage
  - Advantage of soft NoC on top of reconfigurable fabric is that users can customize the NoC to the applications



<Symmetric BFT>



<Asymmetric BFT>