Bubble Sharing: Area and Energy Efficient Adaptive Routers using Centralized Buffers

Syed Minhaj Hassan and Sudhakar Yalamanchili

Center for Research on Experimental Computer Systems
School of Electrical and Computer Engineering
Georgia Institute of Technology

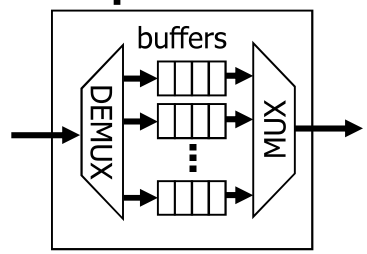
Sponsors: National Science Foundation, Sandia National Laboratories

Overview

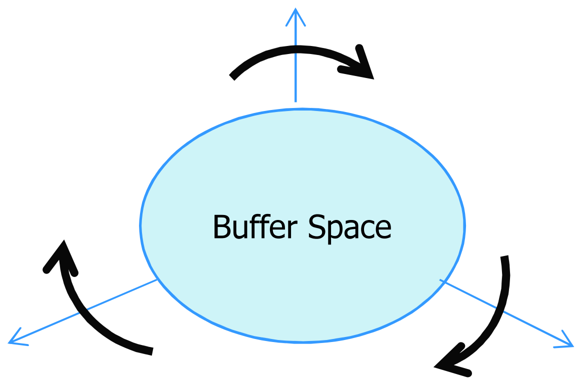
- Buffer Space Reduction Problem
  - Centralized Buffer Router
  - Bubble Flow Control & Its Variants
- Bubble Sharing Flow Control
- Adaptive Bubble Sharing
  - 3 conditions to avoid deadlock
- Results & Conclusion

Router Buffer Space

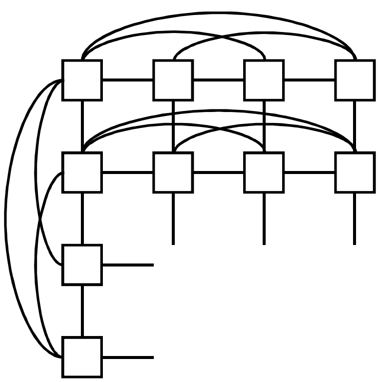
Multiple VCs,
Multiple Virtual Net



•Used for deadlock avoidance / QoS
•64 node mesh: (100 – 400KB)
•Ideal – deadlock avoidance independent of buffer size

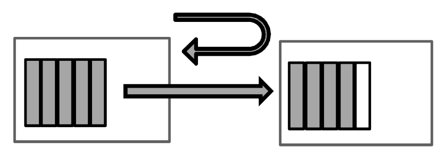


High Radix



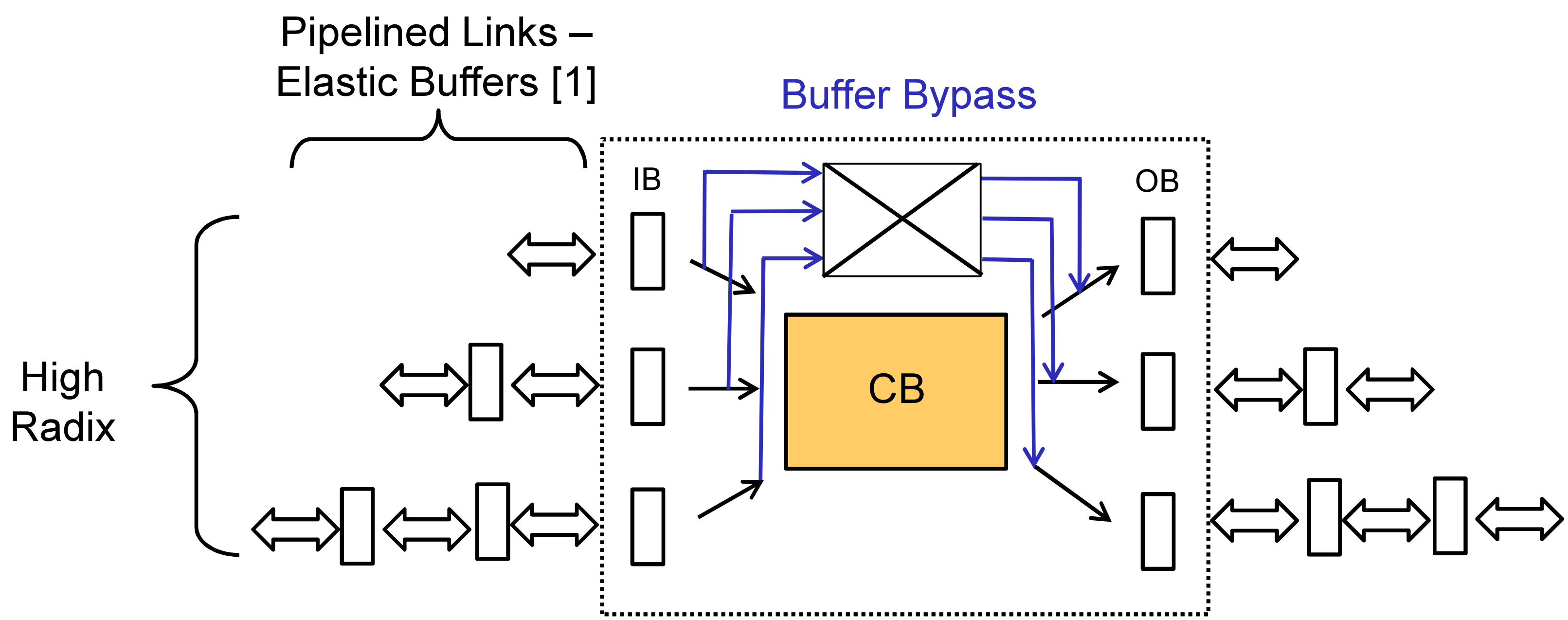
•Reduced hop count
•More wires →↑buffers
•Ideal – buffer space decoupled from radix

Flow Control



•Remove pipeline bubbles & high link utilization
•Buffer size = $\mathcal{F}$(RTT latency)
•Long wires →↑buffers
•Ideal – buffers size decoupled from wire length

Centralized Buffer Routers

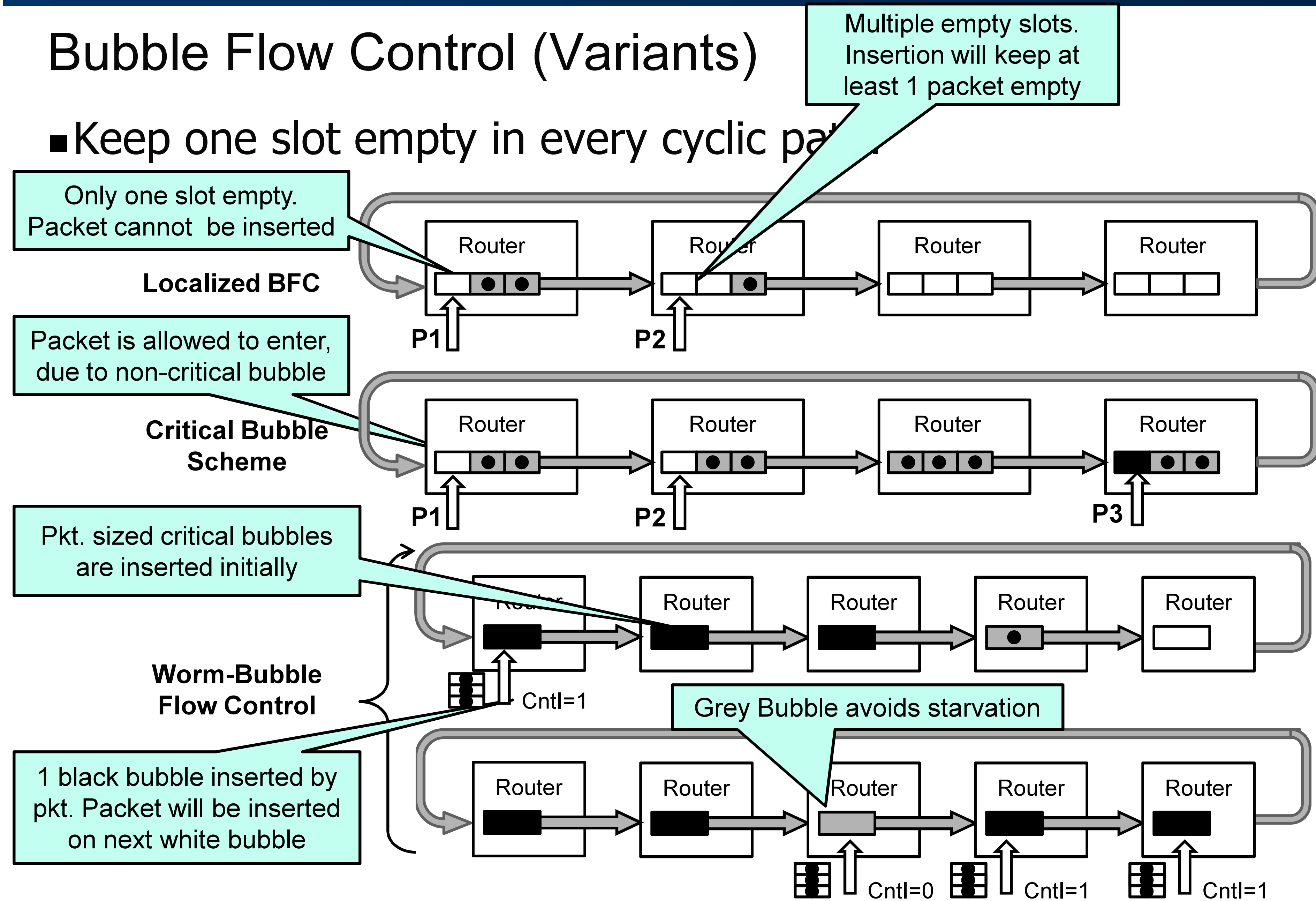


- Central buffers reduce buffer space dependency on radix.
- Elastic Buffer (EB) links to decouple buffer size from wire length.
- Buffer bypass to reduce latency at low load.
- Bubble flow control (Pkt. based) using central buffers for deadlock avoidance **without using VCs**.

[1] Michelogiannakis, G. Elastic Buffer Flow Control for On-Chip Networks, HPCA 2009

Bubble Flow Control (Variants)

- Keep one slot empty in every cyclic path



[1] Lizhong Chen. Worm-Bubble Flow Control, HPCA 2013

Overview

- Need for Buffer Space Reduction
  - Centralized Buffer Router – Overview
  - Bubble Flow Control & Its Variants
- Bubble Sharing Flow Control
- Adaptive Bubble Sharing
  - 3 conditions to avoid deadlock
- Results & Conclusion

Bubble Sharing - I

- Implement WBFC with central buffers.
- Central buffers can be organized as slots of 2-3 flits.
  - Shared pool of worm-bubbles.
  - Multiple can be assigned to each port.

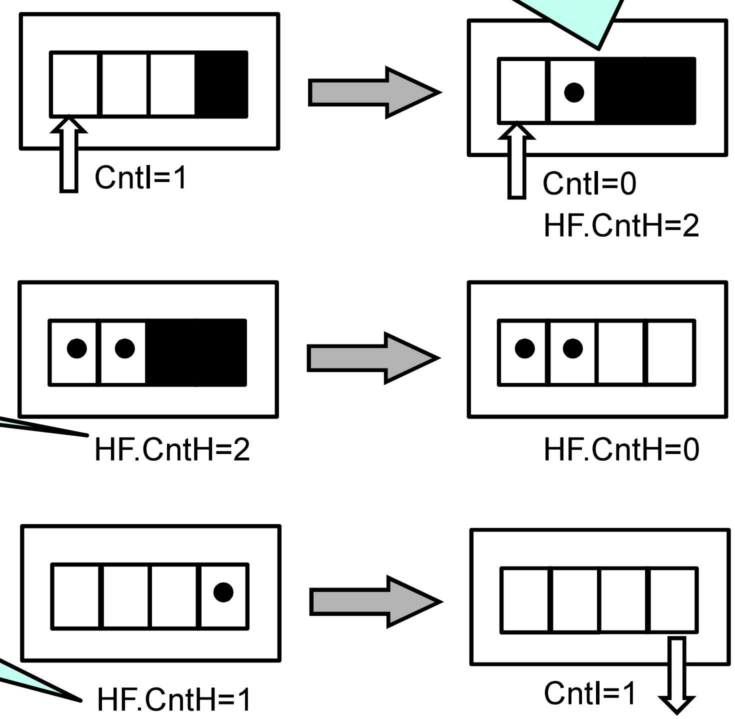
Shared pool allows multiple worms to be made black simultaneously.

- Injection:
  - if (CntI+WhiteBubbleCnt >= PktS_WB)
- Transit:

Marked bubbles are unmarked reducing CntH in head flit.

- Ejection:

Pass remaining count to corresponding ring of ejecting router.

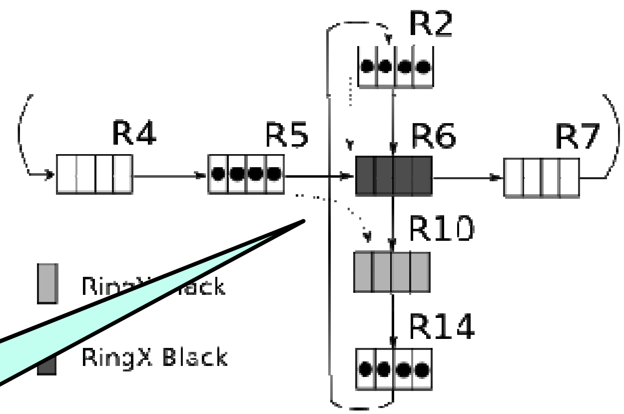


- Grey Bubble: Similar to WBFC.

  Require Backward Displacement

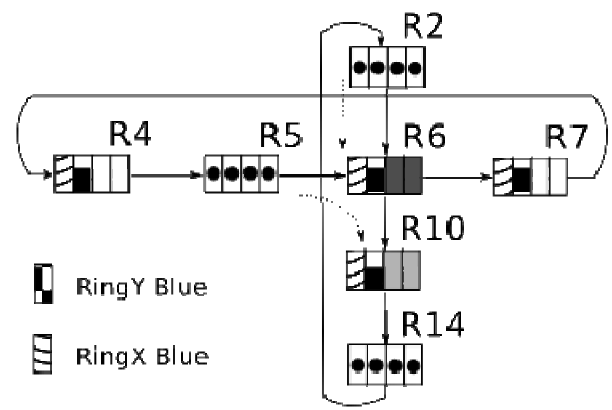
Bubble Sharing - II

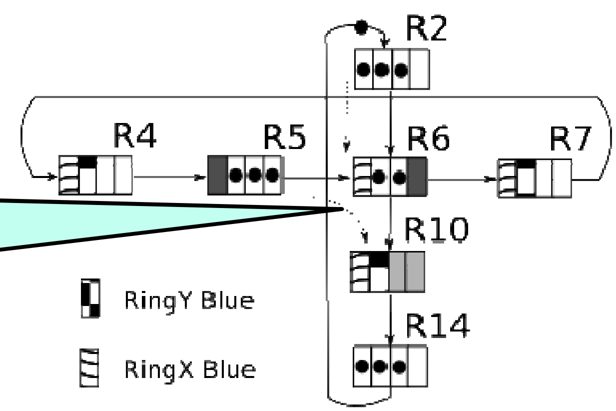
- Sharing may result into 1 ring taking all the bubbles at a particular router, leading to deadlock.



RingX took all of R6.
RingY cannot move.
R5 is stuck as well.

- Introduce **blue bubbles**, 1 dedicated for each ring per router.
  - Act as white bubble for corresponding ring
  - Black bubble for all other rings
  - Ensures at least 1 bubble for each ring





Blue Bubble allows ringY to move forward.
Should be reclaimed immediately after flit traversal.

Bubble Sharing - III

- A packet passes the remaining count at the ejection point.
  - CntI keeps increasing at a particular node
  - All black bubbles are inserted by that node
  - Can lead to starvation of other nodes

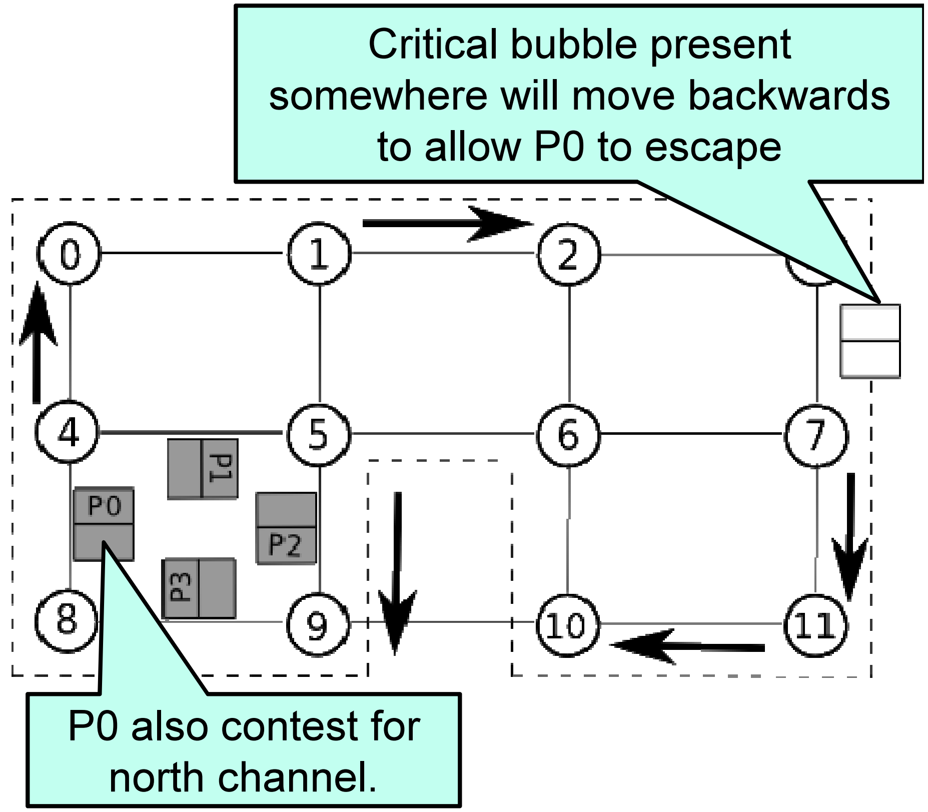
- Solution: Bkwd displacement of CntI
  - If CntI > PktS_WB-1
    - bkwdDisp(CntI)
  - This means routers giving their black bubbles to other routers in the ring

Overview

- Need for Buffer Space Reduction
  - Centralized Buffer Router – Overview
  - Bubble Flow Control & Its Variants
- Bubble Sharing Flow Control
- Adaptive Bubble Sharing
  - 3 conditions to avoid deadlock
- Results & Conclusion

Adaptive Bubble Sharing

- Bubble Coloring Scheme
  - Allow adaptivity by providing a virtual escape ring spanning all routers.
  - Virtual ring is kept deadlock free using CBS (pkt. based).



- Adaptive Bubble Sharing
  - Modify bubble coloring for flit level to be used with CBRs.
  - 3 conditions for deadlock freedom
    1. There must be an escape path from all nodes.
    2. Packets leaving the virtual ring must be consumed.
    3. Every packet should always be able to contest for the escape path.

[1] Wang R. Bubble Coloring: Avoiding Routing- and Protocol-induced Deadlocks With Minimal Virtual Channel Requirement, ICS 2013

Satisfying Condition 1 (There must be an escape path from all nodes)

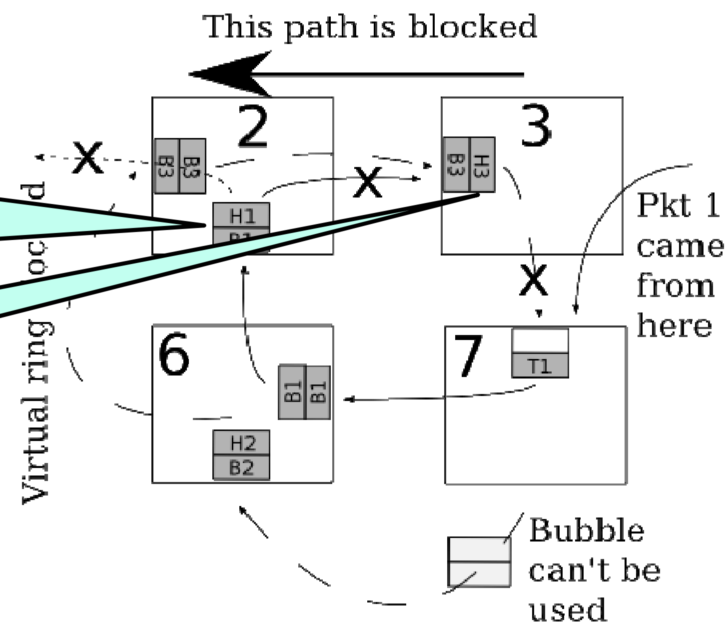
- Virtual ring similar to bubble coloring can be used as an escape path.
  - Use bubble sharing instead of CBS.
- Bubble Coloring allows 180 degree turns.
  - Escape path in opposite direction to the deterministic path.
  - Not possible with flit based wormhole networks.
    - Body & tail flit can remain behind in the previous router.
    - 2 such turns leads to a cycle.
- Solution:
  - Use 2 bubble shared virtual rings going in opposite direction.
    - Prohibit 180 degree turns.
    - Both rings will be deadlock free.

Satisfying Condition 2 (Packets leaving the virtual ring must be consumed)

- Every packet leaving the ring needs to be consumed completely.
  - Not ensured with interacting ring & non-ring channels.

P1 coming from east of node 3.
Ring going west (with router 2 & 3 is blocked.
P1 is distributed in node 2, 6 & 7.
P1 wants to take the escape ring going east.

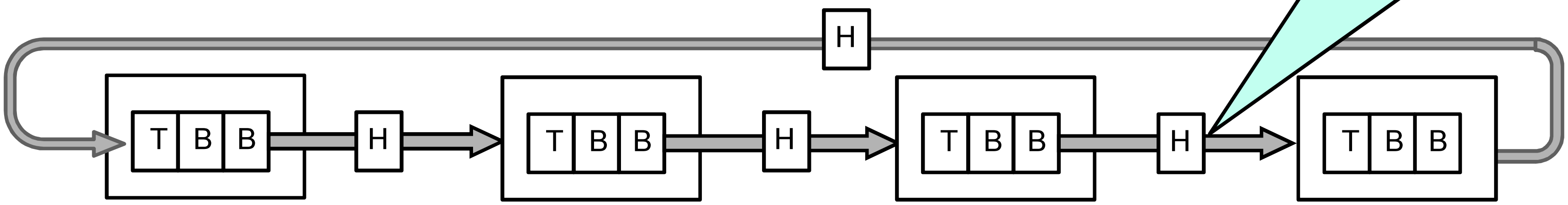
P3 at router 3 wants to move to 7.
Stuck because of tail of P1.
P1 is waiting for P3 to progress. (deadlock)
Bubbles cannot solve this problem.



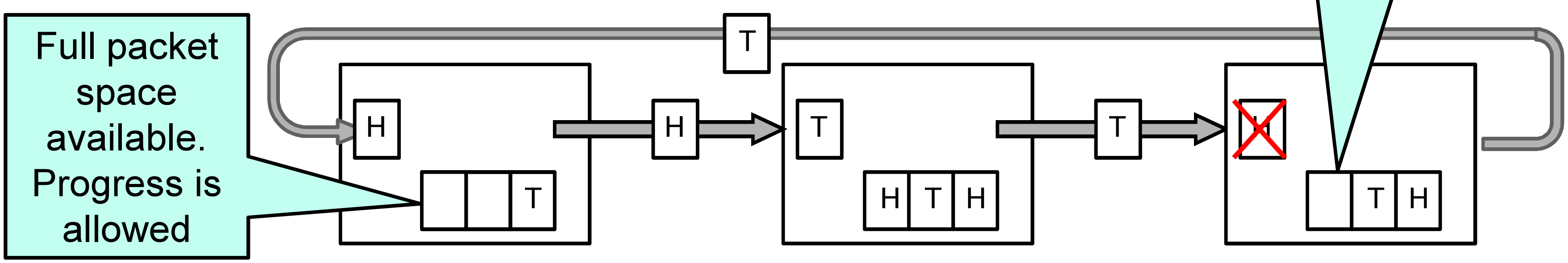
- **Sol:** Check if there is space of a complete packet in the central buffer, before ejecting it from the ring channels.
  - Ensures that when a packet leaves the ring, it is completely drained.

Satisfying Condition 3 (Every packet should always be able to contest for the escape path)

- EB links used in CBRs does not guarantee head flits to not get stuck in link pipelines.



- **Sol:** Use packet based bubble flow control for non-ring channels.
  - Condition 2 is also satisfied by this.
    - Channels used to leave the ring are also non-ring channels



Satisfying Condition 3 (Yellow Bubbles)

- Problem:
  - Channels within the ring is allowed to take more bubbles than non-ring ones. (due to previous limitation).
    - Occupy most of the pool of white bubbles
    - Poor performance of non-ring channels
- Sol:
  - Reserve yellow bubbles for non-ring channels only.
  - Do not allow channels within the ring to occupy all bubbles.
    - Can only take white & their corresponding black bubbles
    - Keeps the non-ring channels away from starvation

Worm-Bubble Coloring

- Adaptive Bubble Sharing with Edge buffer Routers.
  - Credit Based Flow Control
  - No shared pool of worm-bubbles (Use WBFC)

- Three Conditions
  - Escape Path is Available
    - Virtual Ring with WBFC & 2 opposite rings.
    - Prohibit 180 degree turns.
  - Consume Ejecting Packets
    - Provide a small central buffer to be utilized only when the ejection channel gets stuck.
    - If central buffer is in use, new ejection has to wait.
    - Separate buffer space for both rings.
  - Contest Escape Path
    - Send head flits when downstream buffer is empty (full credits)

Overview

- Need for Buffer Space Reduction
  - Centralized Buffer Router – Overview
  - Bubble Flow Control & Its Variants
- Bubble Sharing Flow Control
- Adaptive Bubble Sharing
  - 3 conditions to avoid deadlock
- Results & Conclusions

Simulation Methodology

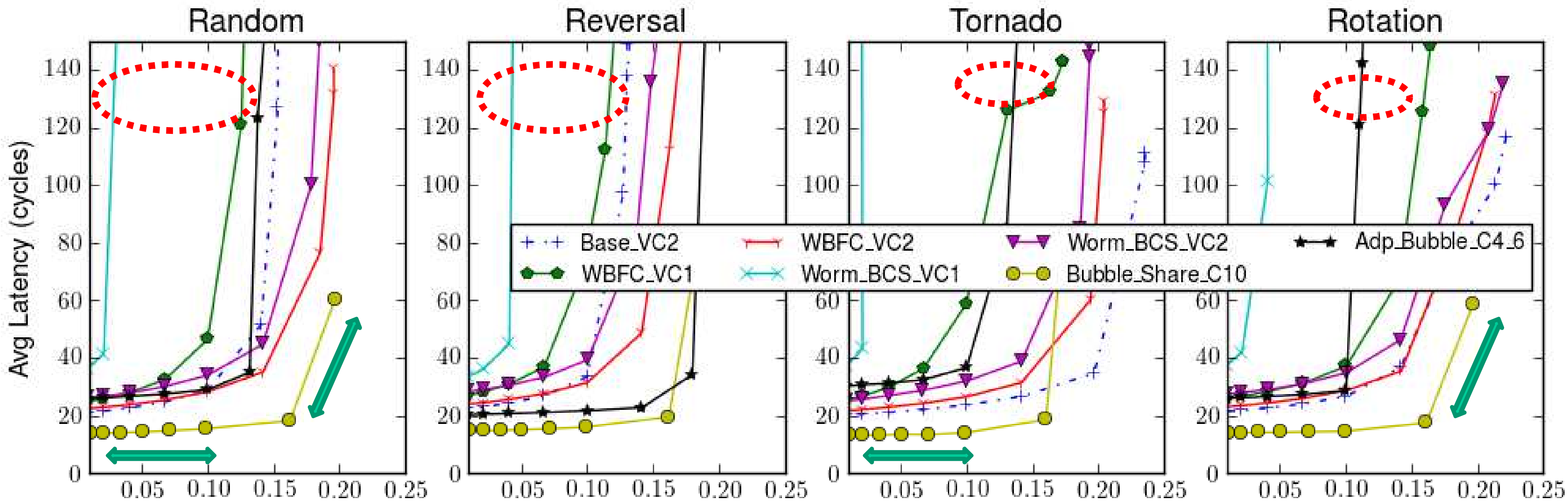
- 5 different routers
  - **Baseline:** Standard 2 stage, multi-VC, 2 flit IB, duato's protocol
  - **WBFC**: Same as baseline, 1 cycle bkwd. displacement.
  - **Worm-BCS**: Same as baseline + 4 flit CB.
  - **Bubble Shared**: (3 black + 1 grey bubbles) per ring + 4 blue bubbles per router + white bubbles = CBx ➔ x+8 flits
  - **Adaptive Bubble Shared**: CBx_y ➔ x-white + y-yellow + 4-blue ➔ x+y+8 flits

- **Edge buffered** routers uses extra VCs for minimal adaptive routing

- **Network**: 4x4 Torus / GHC, 8x8 Torus / GHC
  - GHC has link delay equal to the number of hops between the routers
  - Torus has single cycle link delay

- **Simulations**: 6 flit packets, 128 byte links, 100 million cycles.

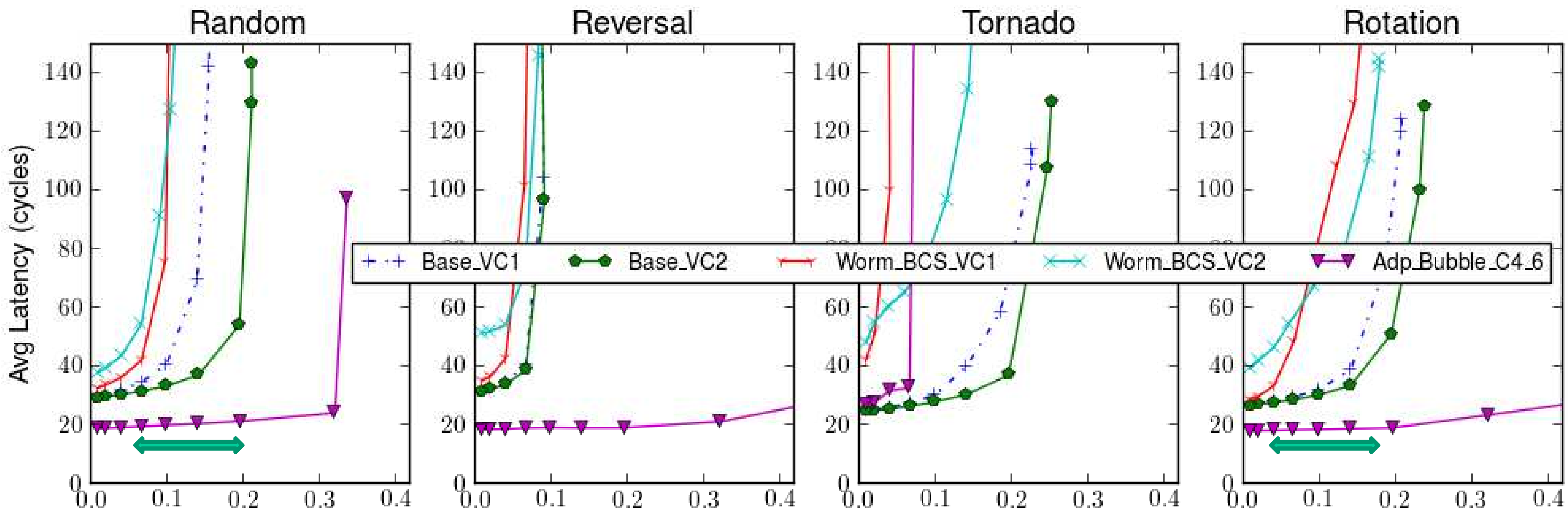
Throughput vs. Avg. Packet Latency (4x4 Torus)



Retired Flits per Node per Cycle vs. Avg Packet Latency (Cycles)

- Single VC solutions with edge buffers has least performance.
- Bubble Sharing has least latency. (Centralized Buffer Router)
- Bubble Sharing has maximum throughput. (Less bubbles)
- Adaptive Bubble Sharing does not perform well (limited number of non-ring channels).

Throughput vs. Avg. Packet Latency (8x8 GHC)



Retired Flits per Node per Cycle vs. Avg Packet Latency (Cycles)

- Adaptive bubble sharing performs significantly better (large number of non-ring channels available)
  - More adaptivity options keeps injection delay low
- Takeaway: 1) Bubble sharing is better for torus (low radix).
  2) Adaptive bubble sharing performs well for GHC (high radix).

Buffer Space Analysis

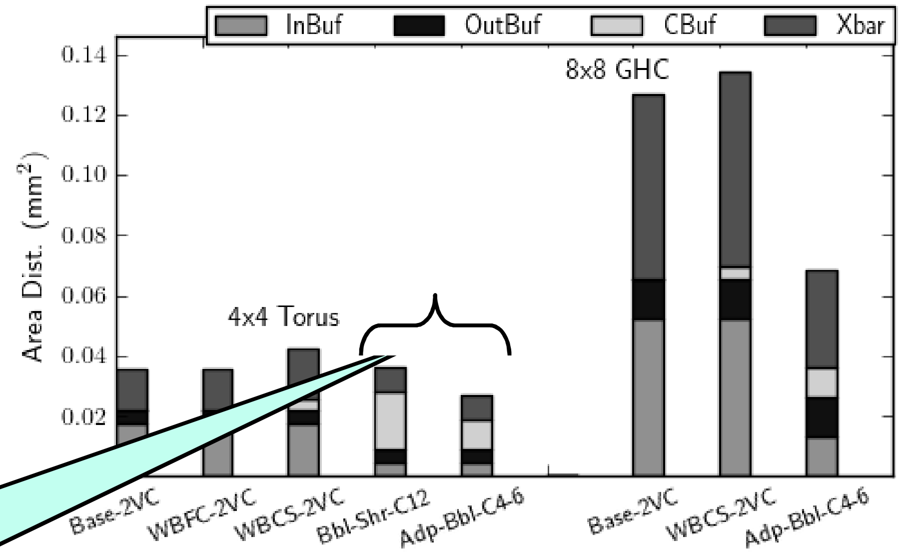
	2D Torus / Router	4x4 GHC / Router	8x8 GHC / Router	
Baseline_VC2	400	560	1200	
WBFC_VC2	400	560	1200	
Worm_BCS_VC2	464	624	1264	
Bubble_Share_C10	448	512	768	CB = 18 flits
Bubble_Share_C12	480	544	800	CB = 20 flits
Adp_Bubble_C4_2	320	384	640	CB = 10 flits
Adp_Bubble_C4_6	384	480	704	CB = 14 flits

2 flit IB / CB Worm, 1 flit OB, 128 bit flits. No msg. class. Blue bubbles are additional.

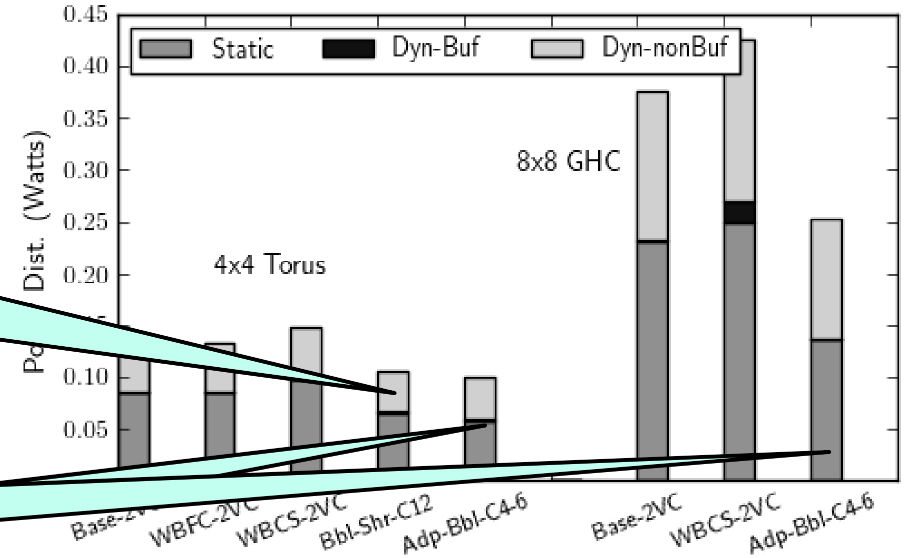
- Edge buffer routers has IB size = $\mathcal{F}$(RTT latency) . CBRs = 1 flit IB.
- Significant reduction for high radix routers with longer links (e.g. 8x8 GHC).
- Rings in x*y Torus = 2x+2y ➔Dedicated Slots / ring = 3 black + 1 grey + 4 blue.
- With 1 white bubble per router, **minimum CB size = 18 and 12 flits for 4x4 and 8x8 Torus**.
- With Adaptive bubble sharing and 2 rings, minimum size reduces to **8 flits**.

Area / Power

- Orion 2.0 is used
  - Activity estimated using timing simulations and fed to Orion
  - Modifications to cater for extra area / power in EB links and arbiters.



1) Input buffer has least area (single VC, single flit).
2) CB takes significant area
3) Crossbar area is also low due to 1 VC.



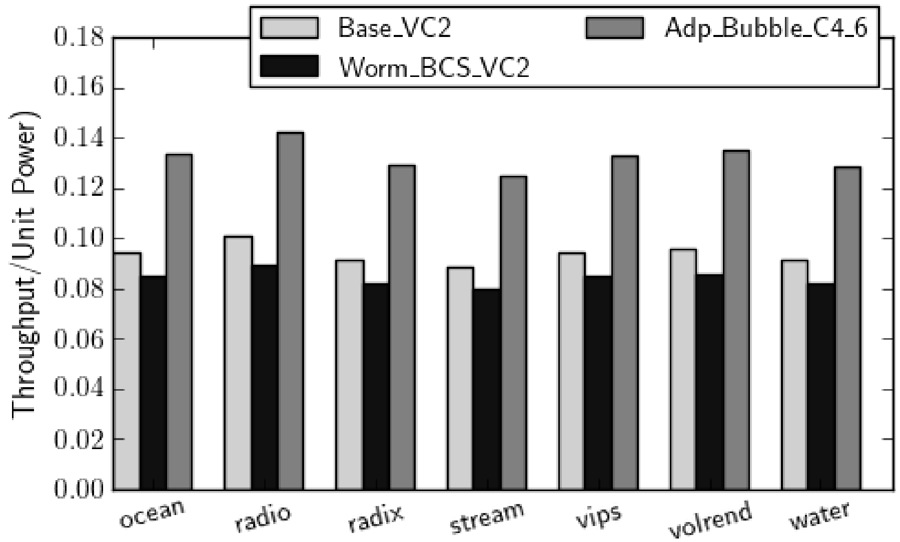
Low Load Power

- Static power for bubble shared router is 24% lower than baseline for 4x4 Torus. (Smaller Crossbar)

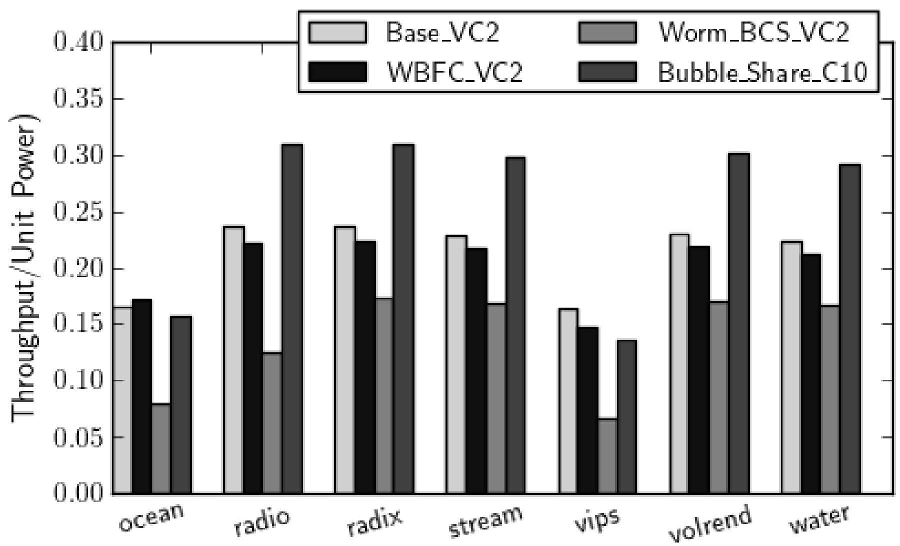
- Adaptive Bubble Shared router reduces it by 32% and 41%.

Results with Real Benchmarks

- With GHC, Adaptive bubble sharing performs the best.



- With Torus, Bubble Sharing surpasses all others.



Conclusions & Next Step

- Proposes variants of bubble flow control in centralized buffer routers.
  - Both deterministic and adaptive.
  - Deterministic version is good for low radix.
  - Adaptive works well for high radix routers.
  - Use less buffering, lower power and higher throughput.

- Next Steps
  - Hardware Implementation
  - Separation of flows to provide bandwidth guarantees with different message types.
  - QoS support in general.
  - Implement CBRs with extremely high radix topology.

THANK YOU !!