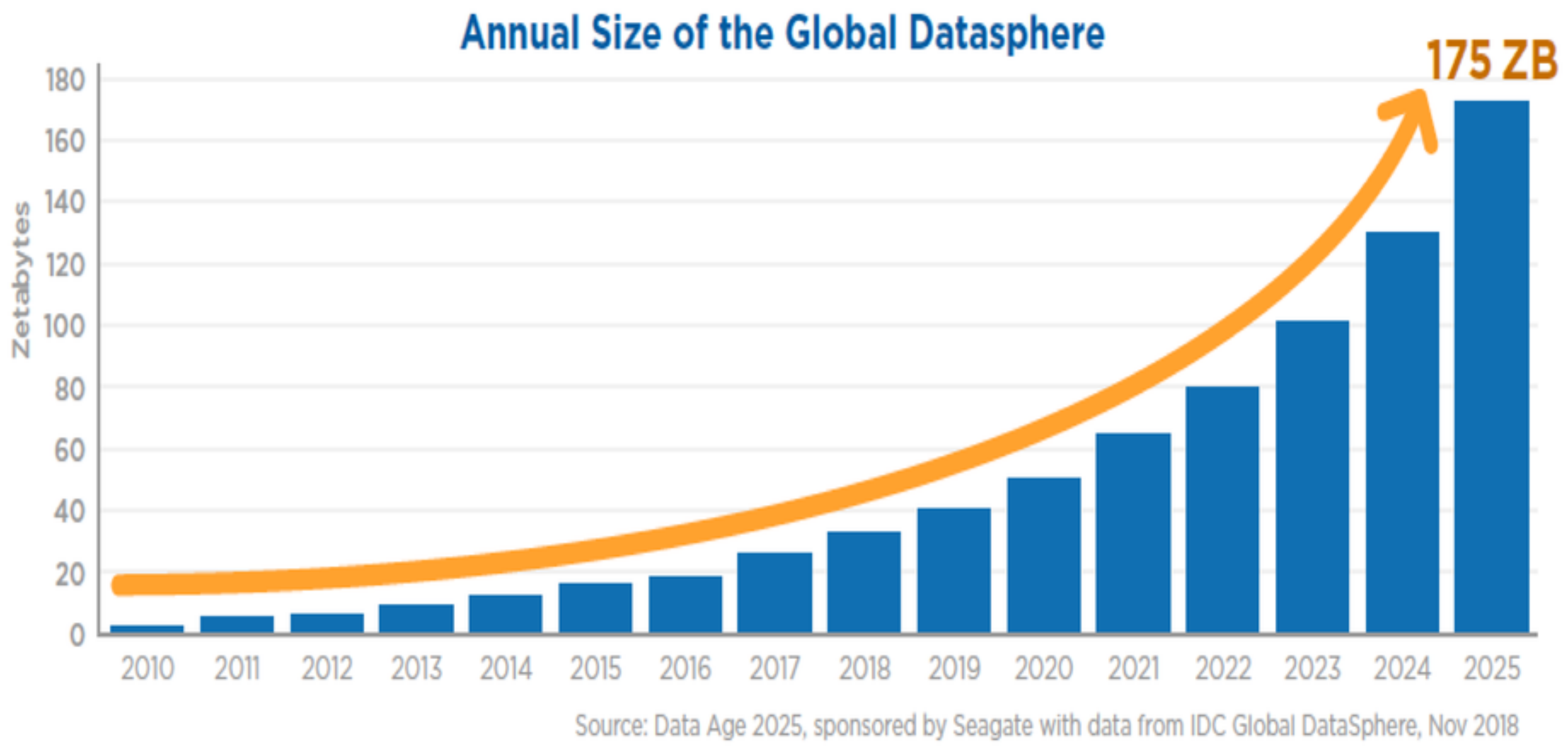
Communication Algorithm-Architecture Co-Design for Distributed Deep Learning

Jiayi Huang Pritam Majumder Sungkeun Kim
Abdullah Muzahid Ki Hwan Yum EJ Kim

UC Santa Barbara (work done at TAMU) Texas A&M University

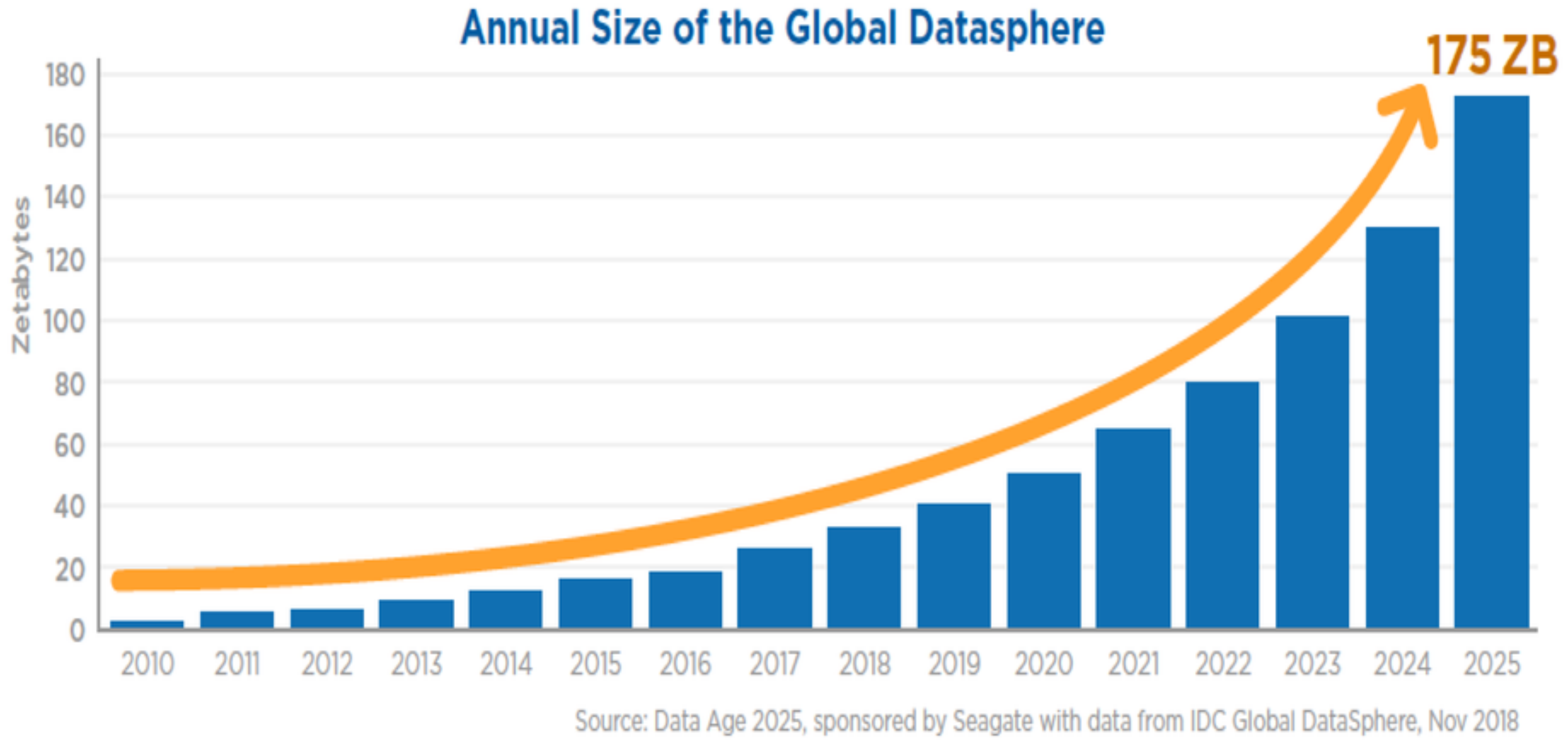
Increasing Demand for Distributed Deep Learning

- Dataset and model sizes are big and increasing



Increasing Demand for Distributed Deep Learning

- Dataset and model sizes are big and increasing



- Train ImageNet in 1 hour – 256 GPUs [Goyal+ 2017]
- AlphaZero – 5000 TPU v1 for games, 64 TPU v2 for training [Silver+ Science'18]

Data-Parallel Training – All-Reduce

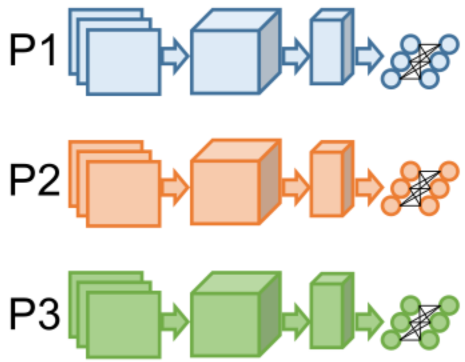
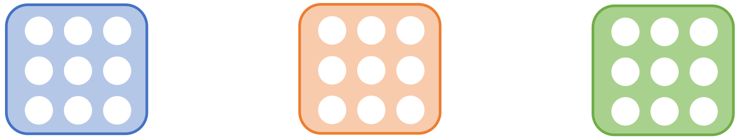


Figure source: Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019



Data-Parallel Training – All-Reduce

Forward

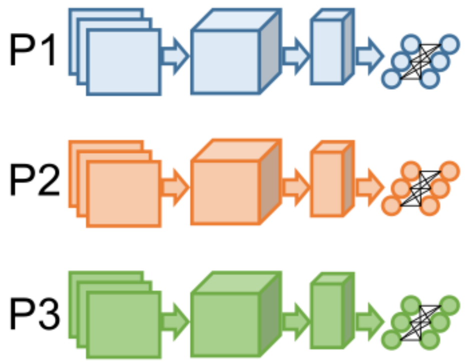
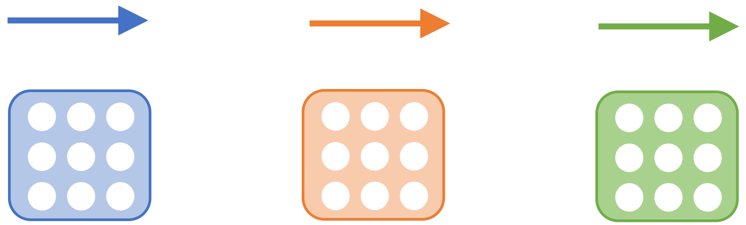


Figure source: Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019



Data-Parallel Training – All-Reduce

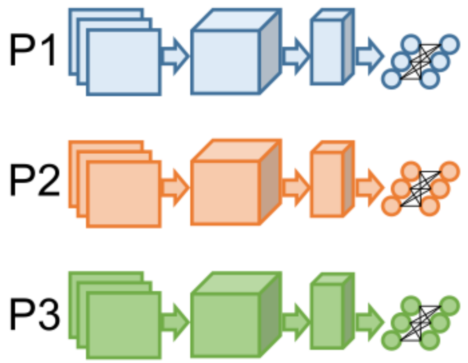
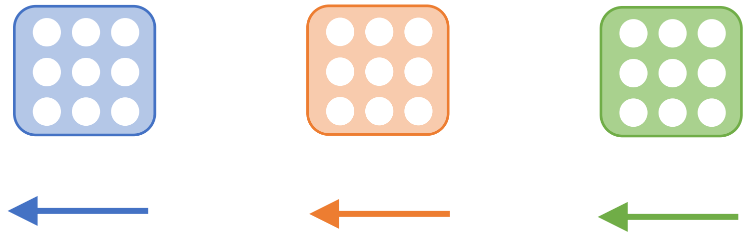


Figure source: Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

Back-Propagation



Data-Parallel Training – All-Reduce

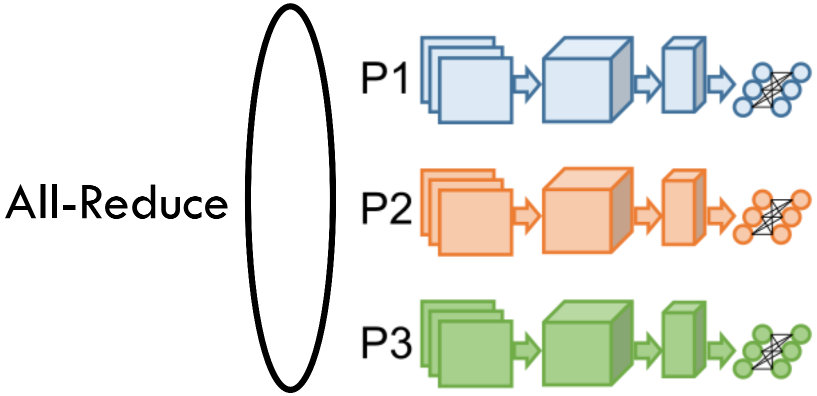
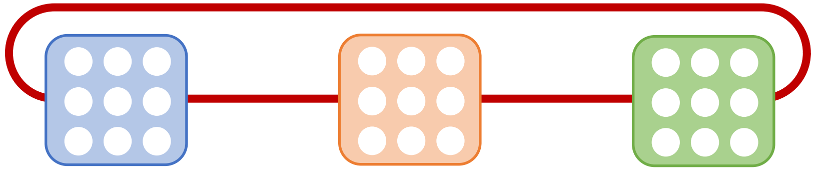


Figure source: Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019



Data-Parallel Training – All-Reduce

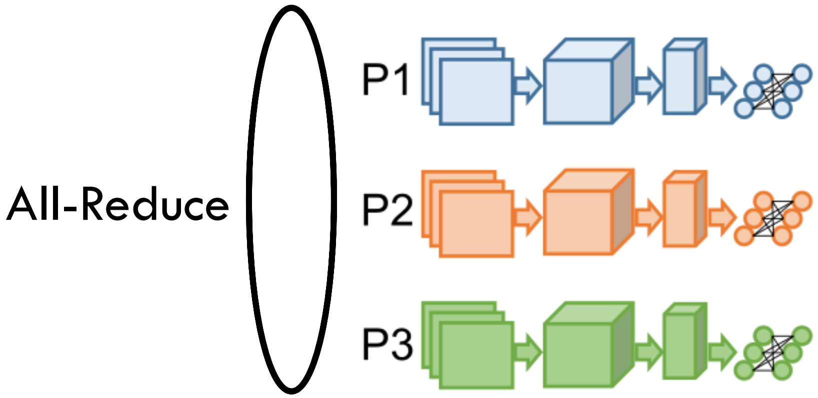
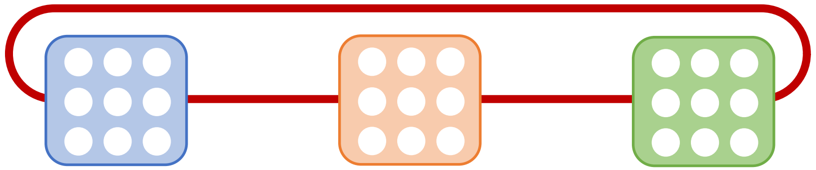


Figure source: Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019



Algorithms	(Small data) Latency	(Large data) Bandwidth-optimal	(Large data) Contention-free	Applied Well on Various Topologies
Ring [Patarasuk+Yuan JPDC'09]	high	✓	✓	✓
Double binary tree [Sanders+ JPC'09]	low	✓	✗	✗ (Topology-oblivious)
2D-Ring [Ying+ NeurIPsW'18]	low	✗	✓	✗ (2D Torus/Mesh)
HDRM [Dong+ HPCA'20]	low	✓	✓	✗ (BiGraph)

Data-Parallel Training – All-Reduce

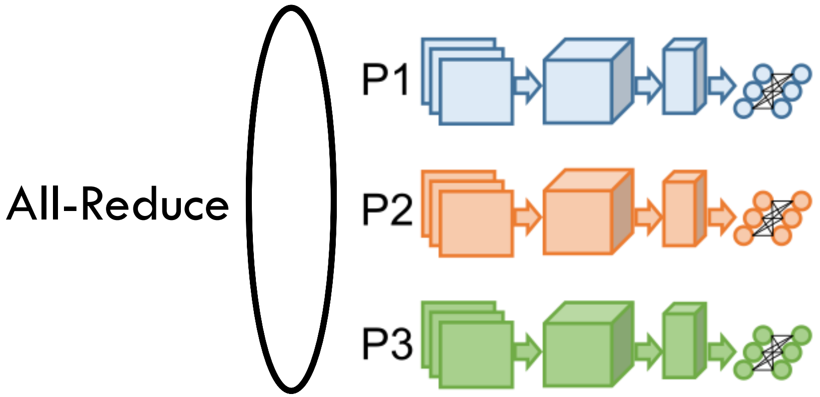


Figure source: Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

MultiTree: Algorithm-Architecture Co-Design

- Topology-aware All-Reduce
- Hardware All-Reduce Scheduling
- Big Message Flow Control

Algorithms	(Small data) Latency	(Large data) Bandwidth-optimal	(Large data) Contention-free	Applied Well on Various Topologies
Ring [Patarasuk+Yuan JPDC'09]	high	✓	✓	✓
Double binary tree [Sanders+ JPC'09]	low	✓	✕	✕ (Topology-oblivious)
2D-Ring [Ying+ NeurIPsW'18]	low	✕	✓	✕ (2D Torus/Mesh)
HDRM [Dong+ HPCA'20]	low	✓	✓	✕ (BiGraph)

Data-Parallel Training – All-Reduce

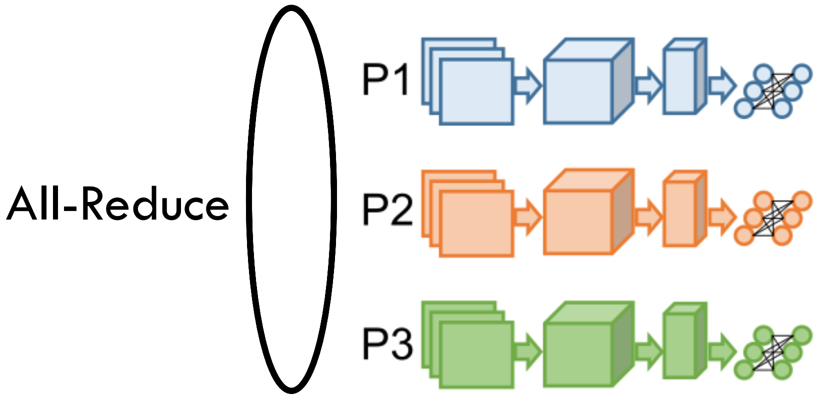


Figure source: Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

MultiTree: Algorithm-Architecture Co-Design

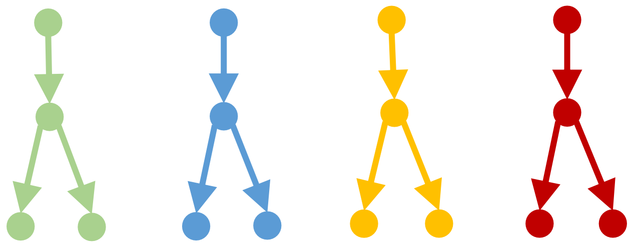
- Topology-aware All-Reduce
- Hardware All-Reduce Scheduling
- Big Message Flow Control

Algorithms	(Small data) Latency	(Large data) Bandwidth-optimal	(Large data) Contention-free	Applied Well on Various Topologies
Ring [Patarasuk+Yuan JPDC'09]	high	✓	✓	✓
Double binary tree [Sanders+ JPC'09]	low	✓	✗	✗ (Topology-oblivious)
2D-Ring [Ying+ NeurIPsW'18]	low	✗	✓	✗ (2D Torus/Mesh)
HDRM [Dong+ HPCA'20]	low	✓	✓	✗ (BiGraph)
MultiTree (Ours)	low	✓	✓	✓

Topology-aware MultiTree All-Reduce

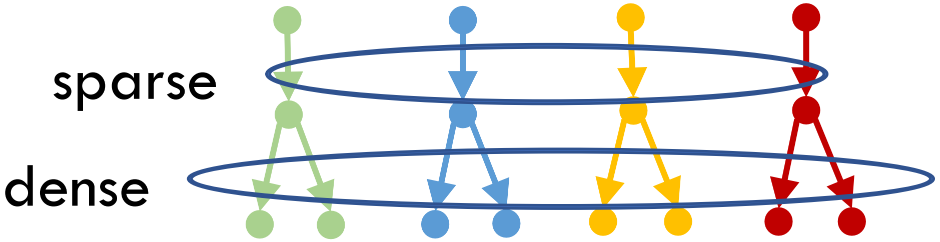
Topology-aware MultiTree All-Reduce

- Topology-aware spanning trees instead of rings
- Combine tree constructions with message scheduling – Contention-free



Topology-aware MultiTree All-Reduce

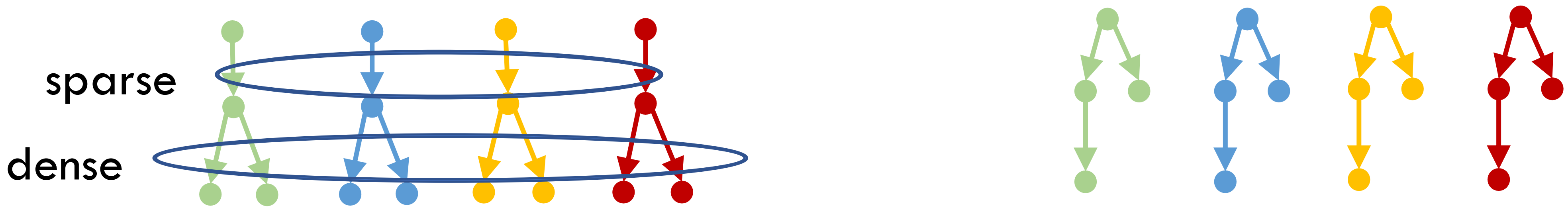
- Topology-aware spanning trees instead of rings
- Combine tree constructions with message scheduling – Contention-free



- Insight: *tree levels closer to leaves are denser than tree levels closer to roots*

Topology-aware MultiTree All-Reduce

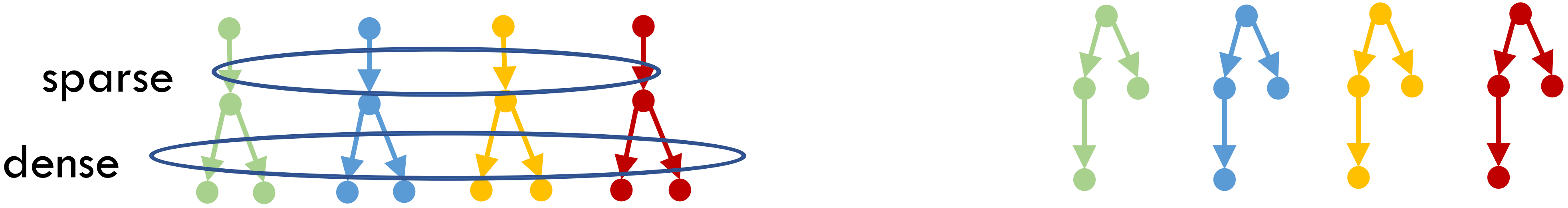
- Topology-aware spanning trees instead of rings
- Combine tree constructions with message scheduling – Contention-free



- Insight: *tree levels closer to leaves are denser than tree levels closer to roots*
- Top-down approach – move more communications closer to roots

Topology-aware MultiTree All-Reduce

- Topology-aware spanning trees instead of rings
- Combine tree constructions with message scheduling – Contention-free

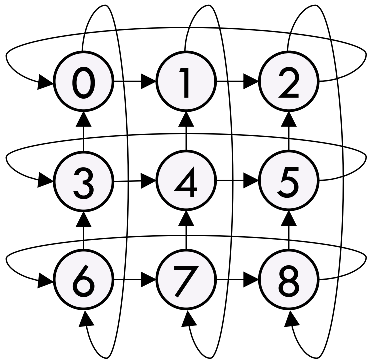


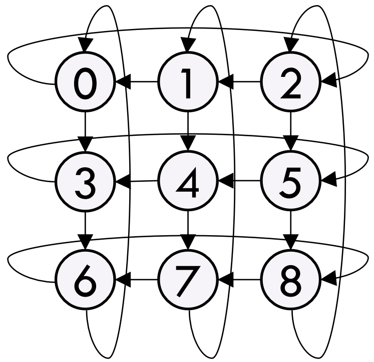
- Insight: *tree levels closer to leaves are denser than tree levels closer to roots*
- Top-down approach – move more communications closer to roots

- Constructing trees – link allocation problem (global coordination)
  - Allocate link for each step (level) to build the trees progressively

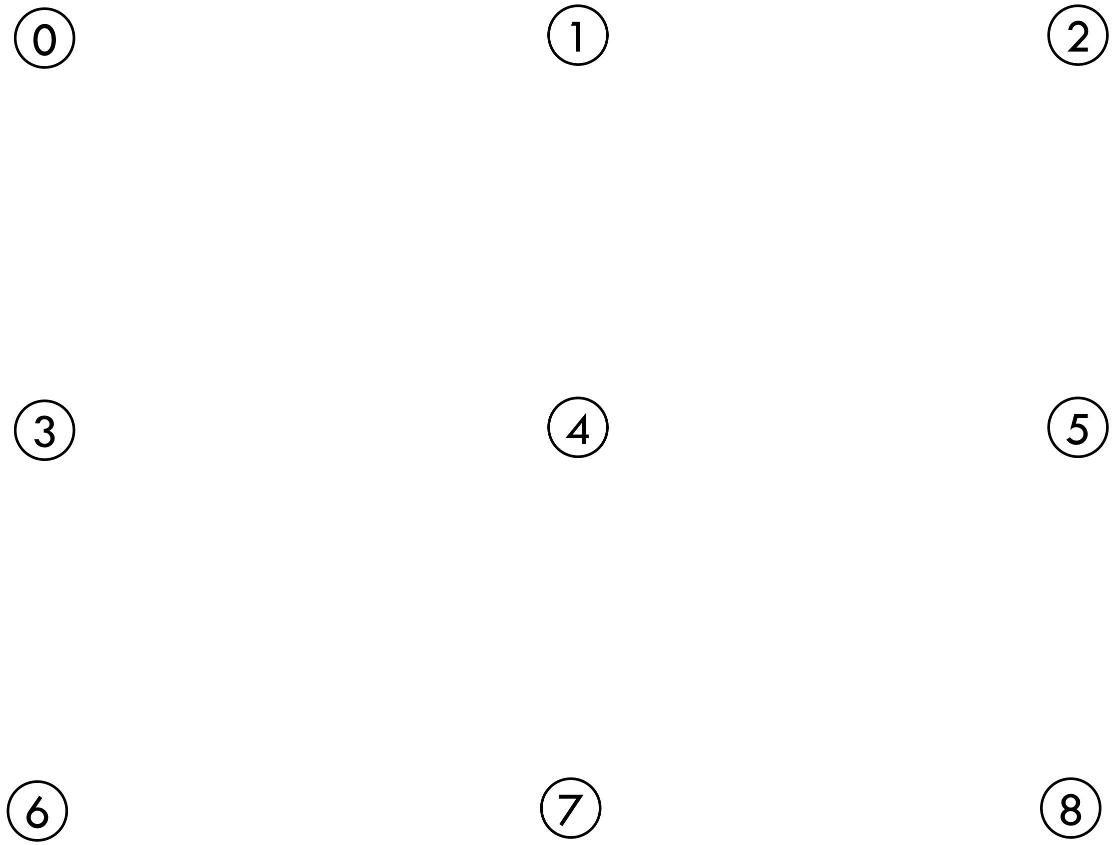
Multitree Example (Time Step 1)

X+, Y+ Links

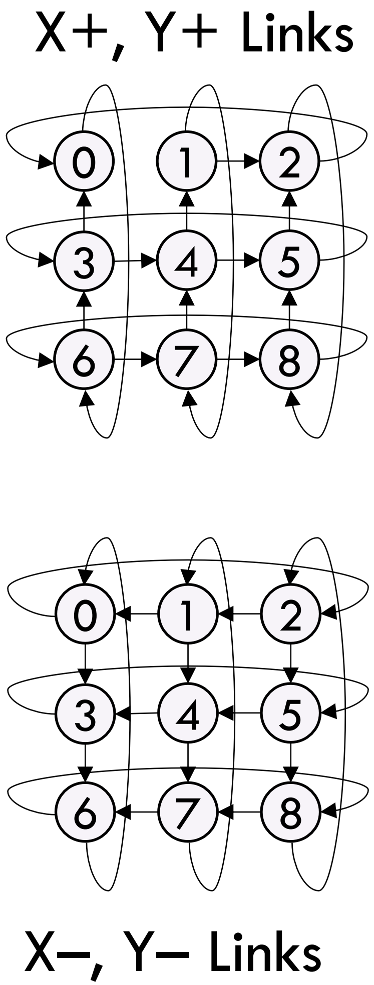


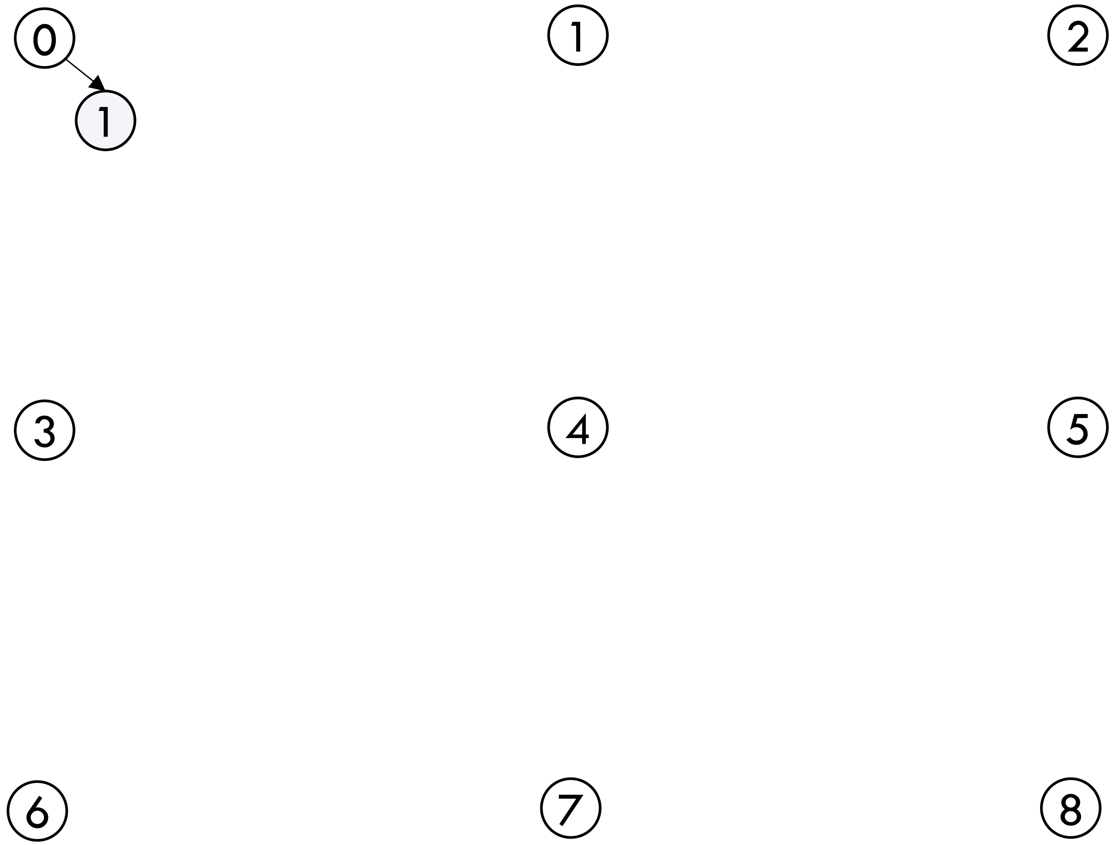


X–, Y– Links



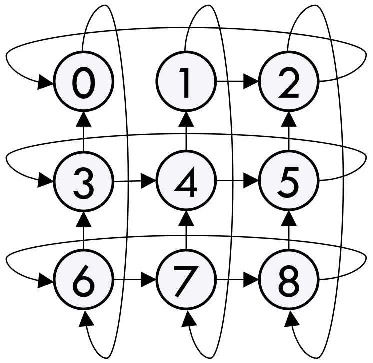
Multitree Example (Time Step 1)

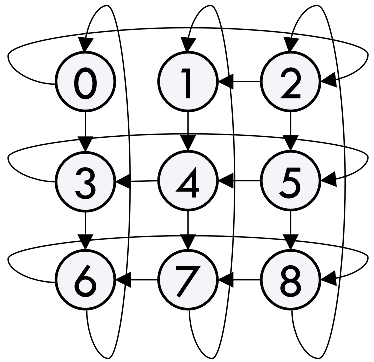




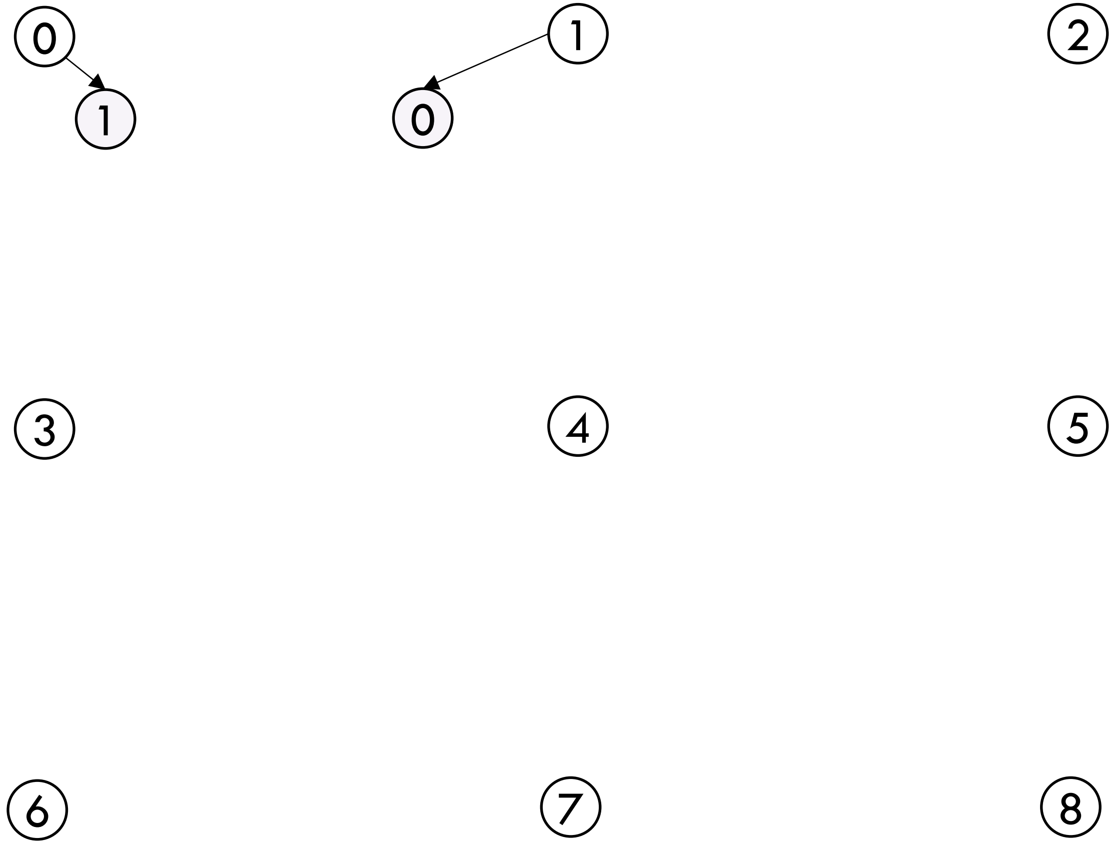
Multitree Example (Time Step 1)

X+, Y+ Links

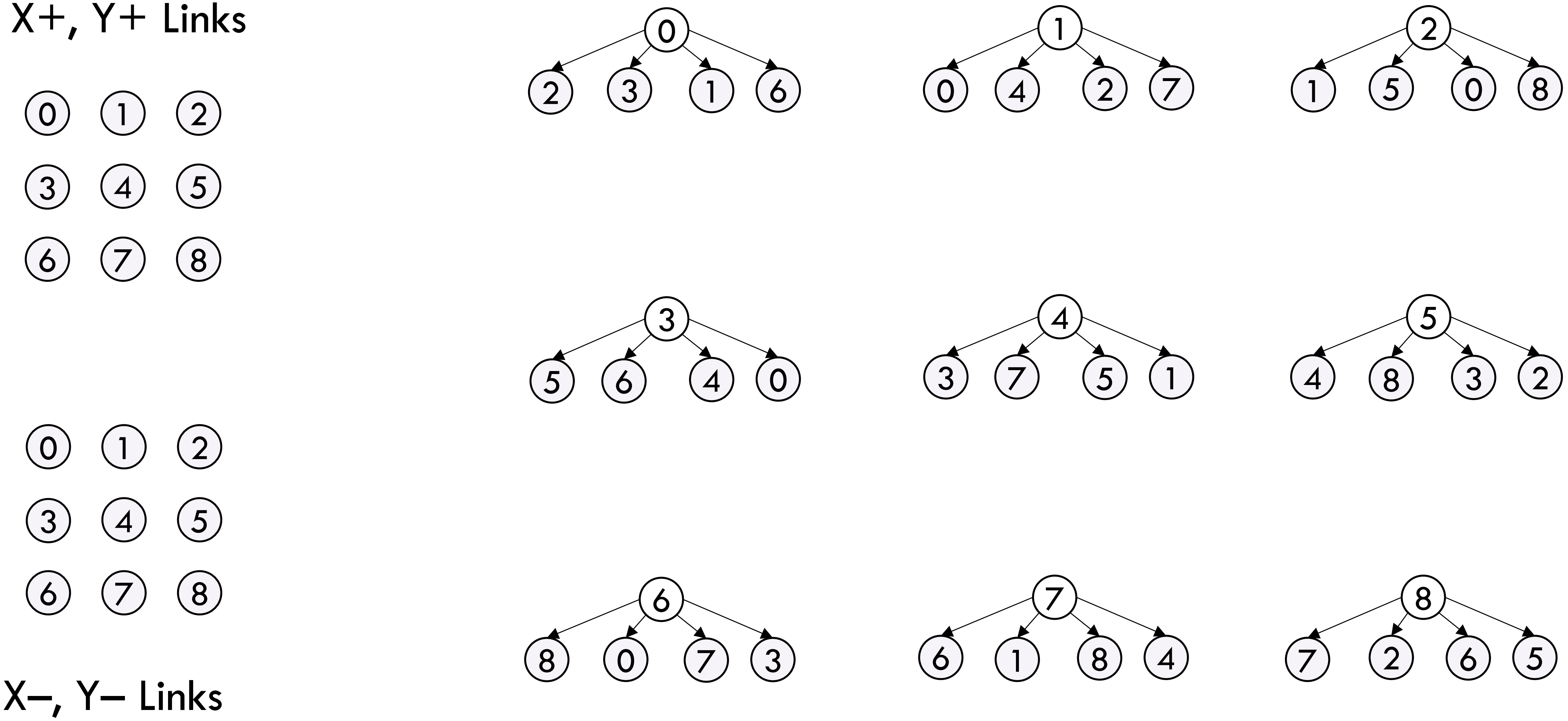




X–, Y– Links

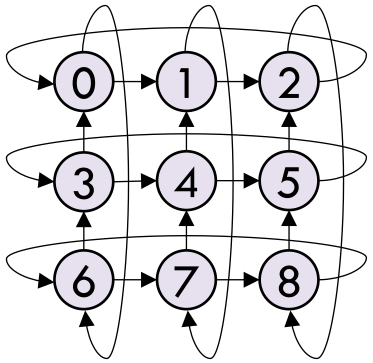


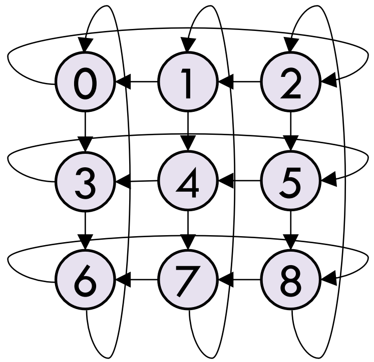
Multitree Example (Time Step 1)



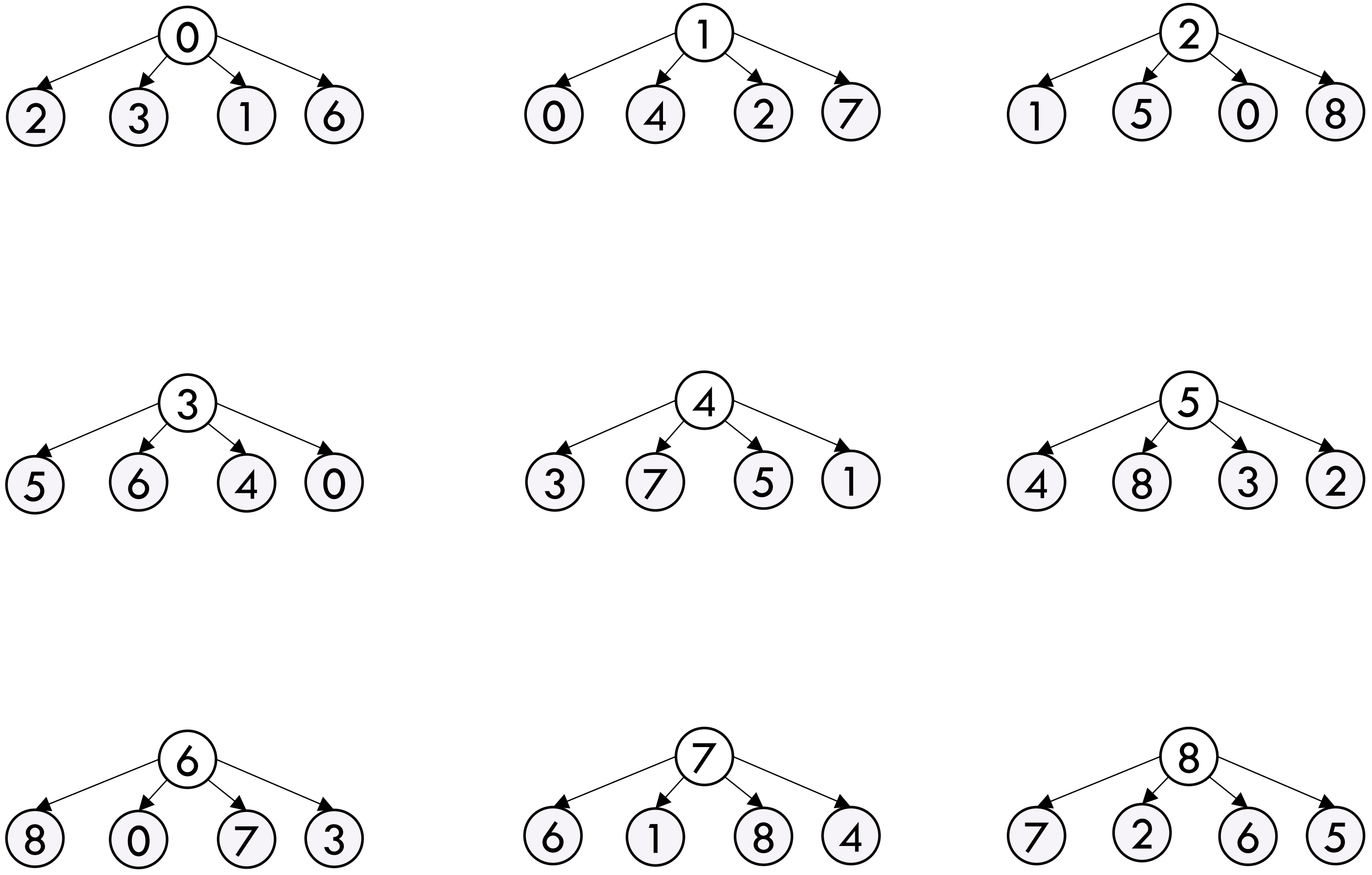
Multitree Example (Time Step 2)

X+, Y+ Links

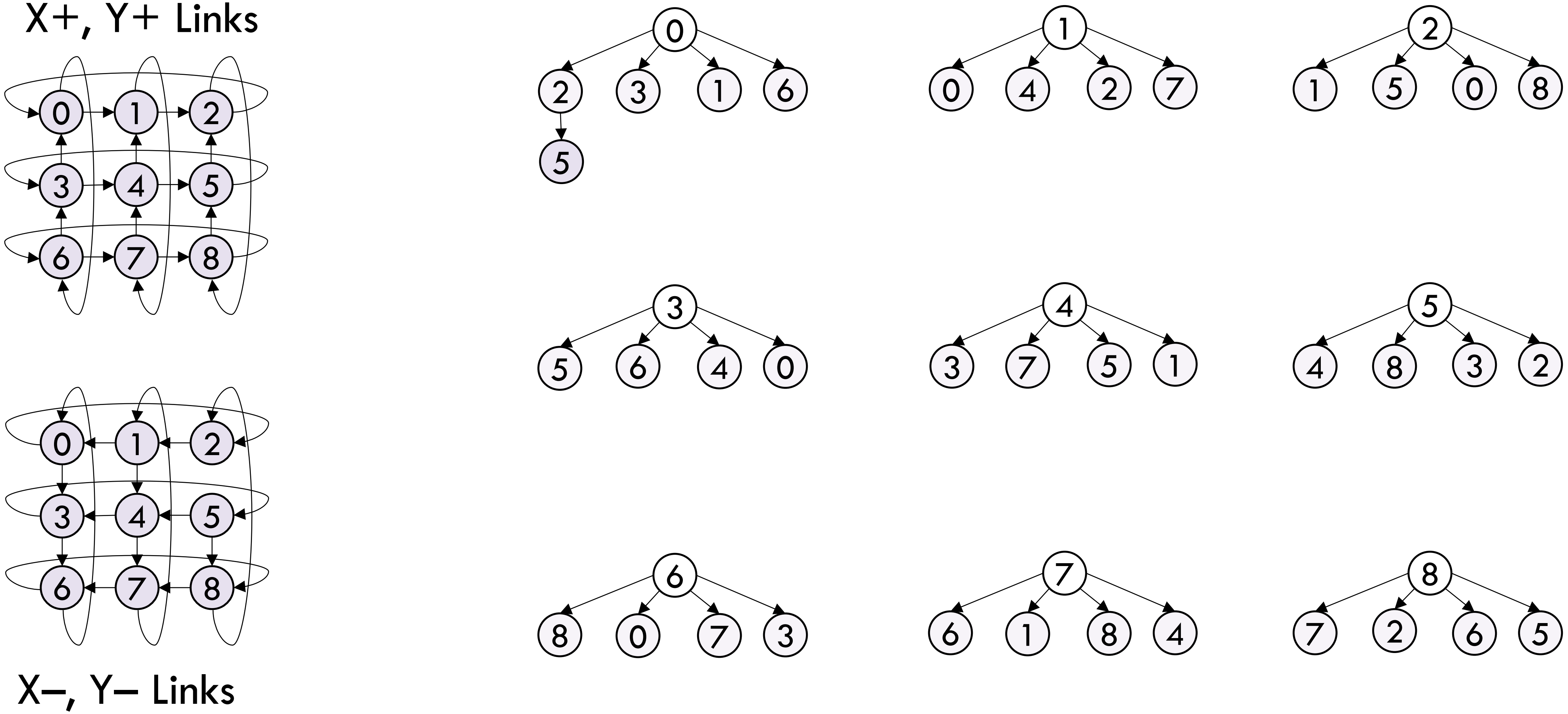




X–, Y– Links

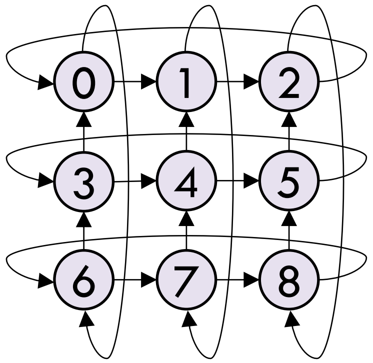


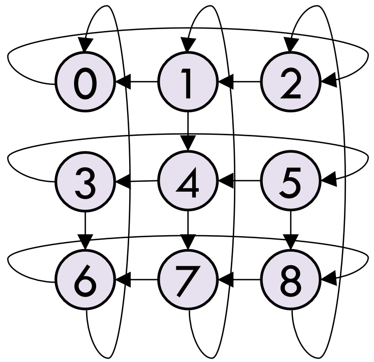
Multitree Example (Time Step 2)



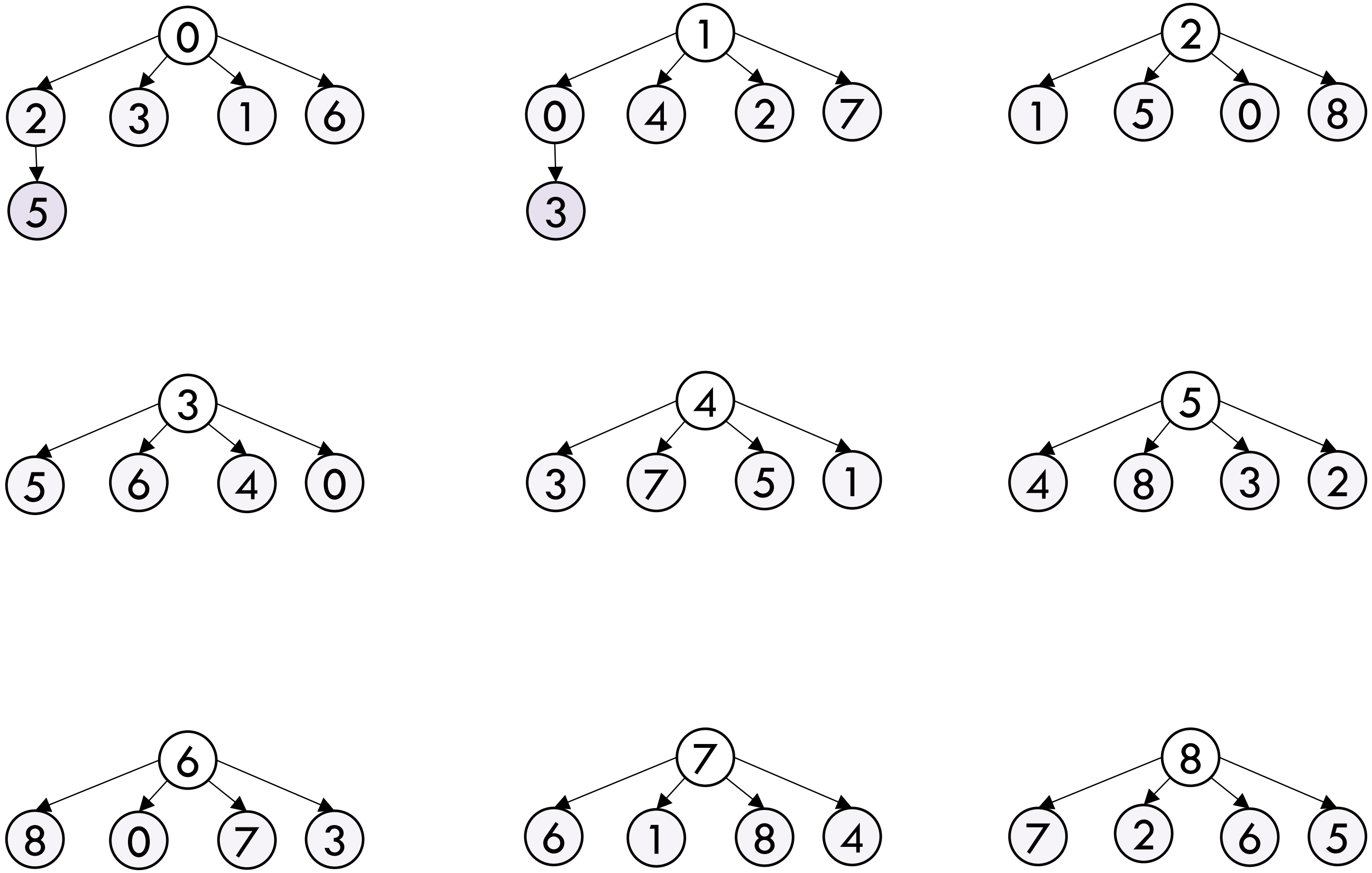
Multitree Example (Time Step 2)

X+, Y+ Links

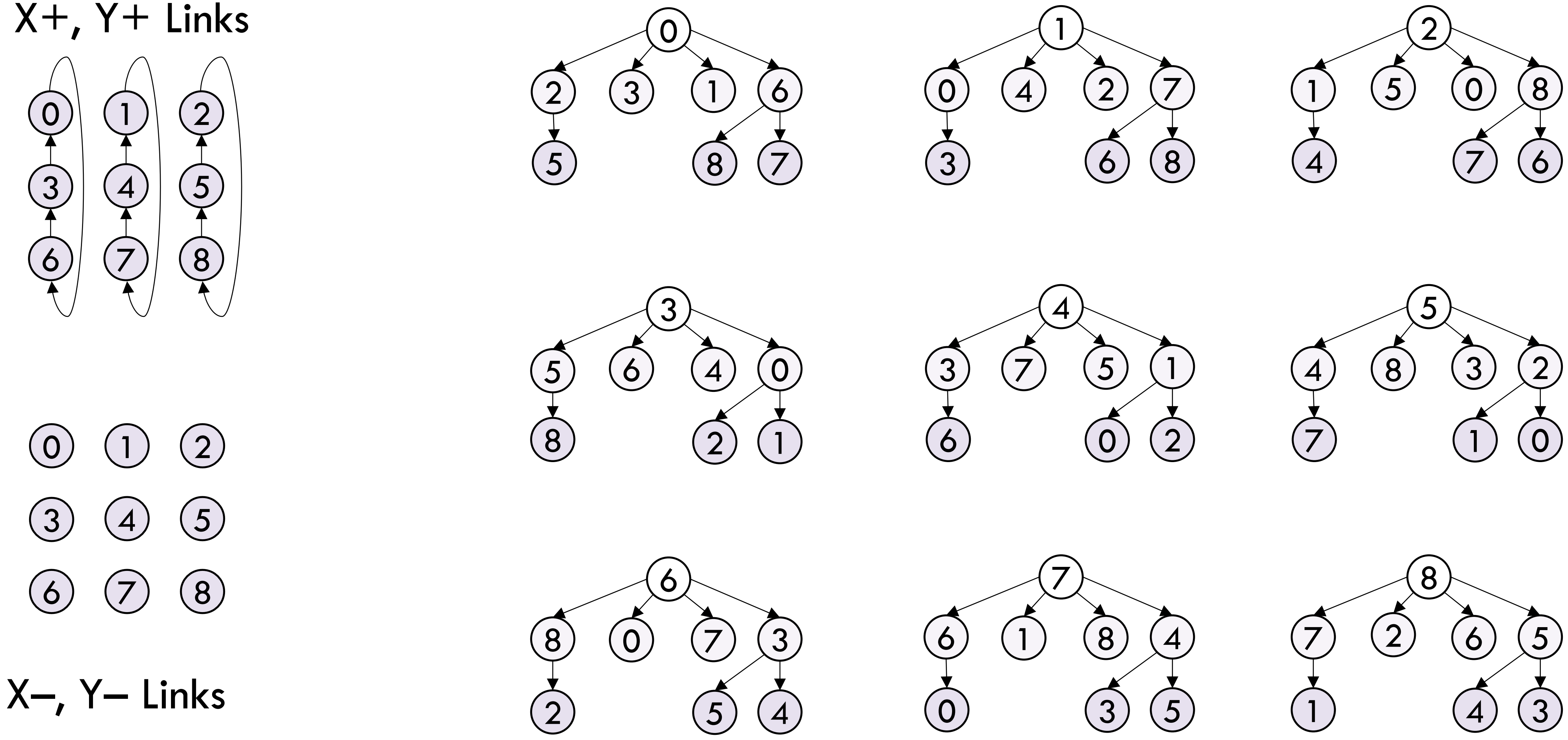




X–, Y– Links

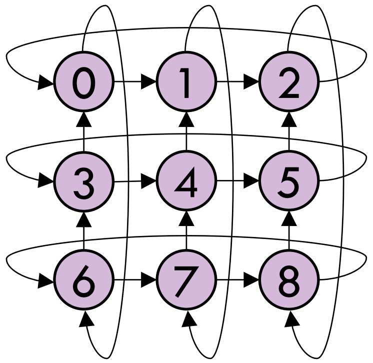


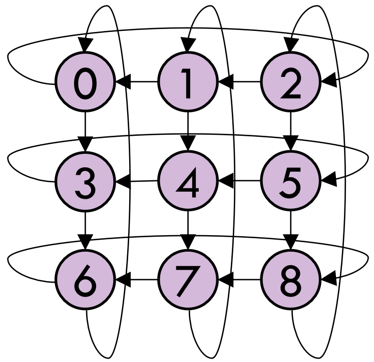
Multitree Example (Time Step 2)



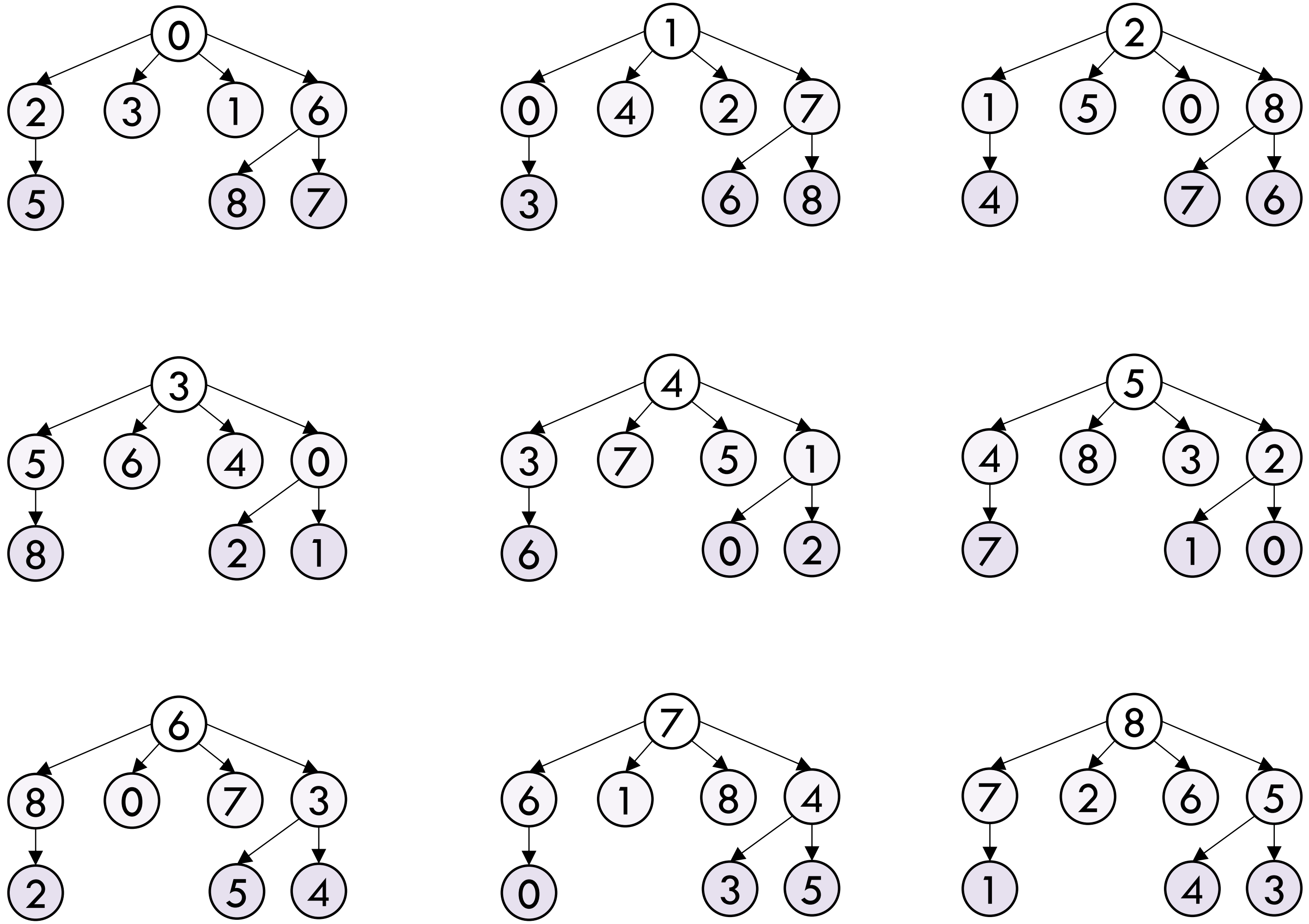
Multitree Example (Time Step 3)

X+, Y+ Links



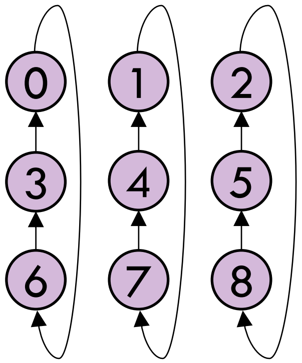


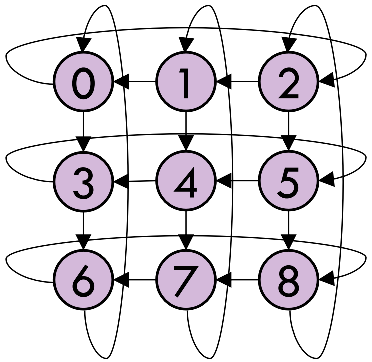
X–, Y– Links



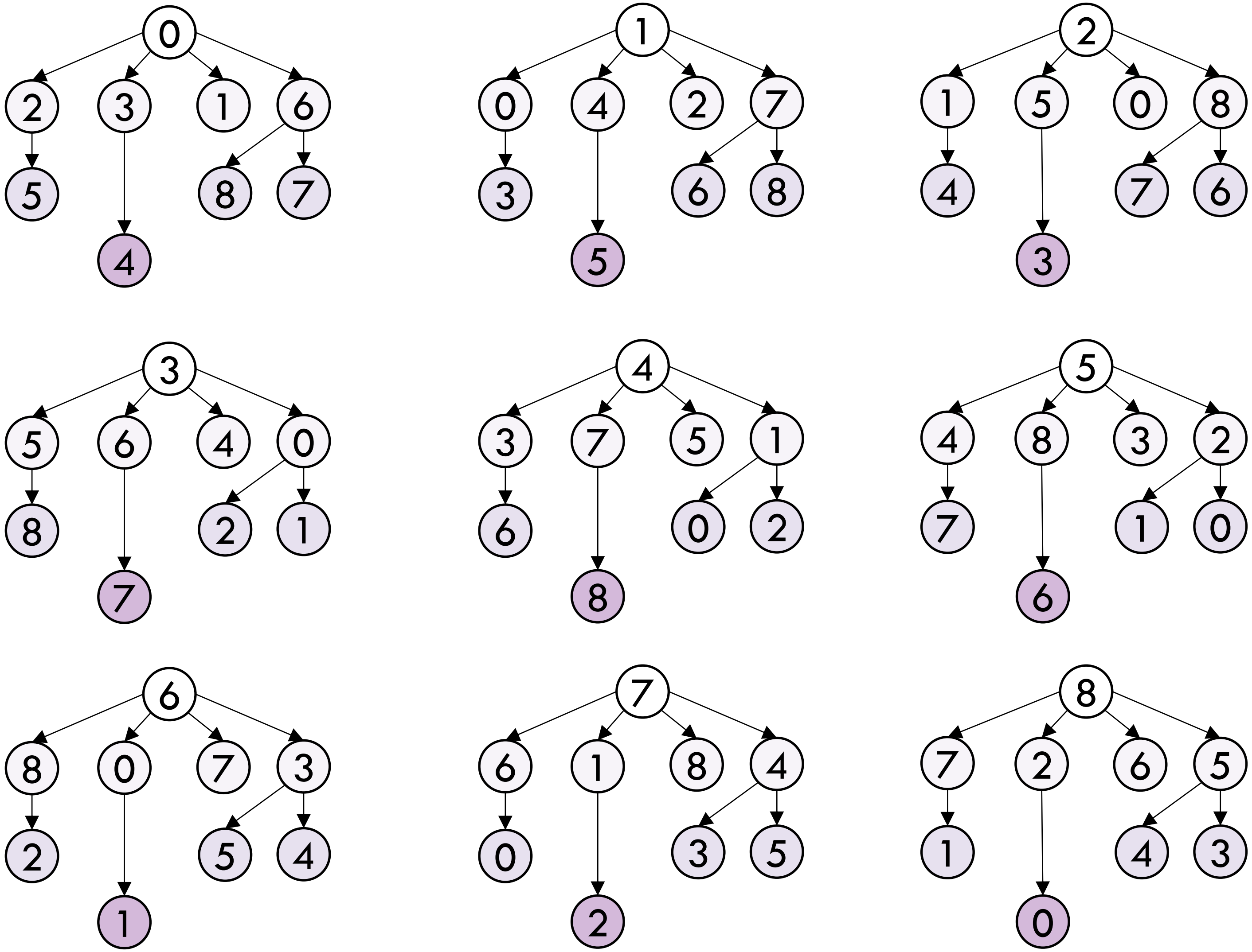
Multitree Example (Time Step 3)

X+, Y+ Links



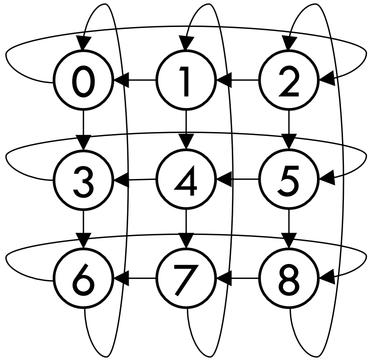


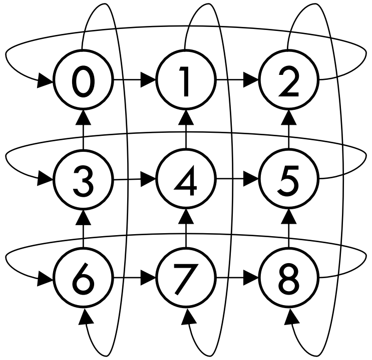
X–, Y– Links



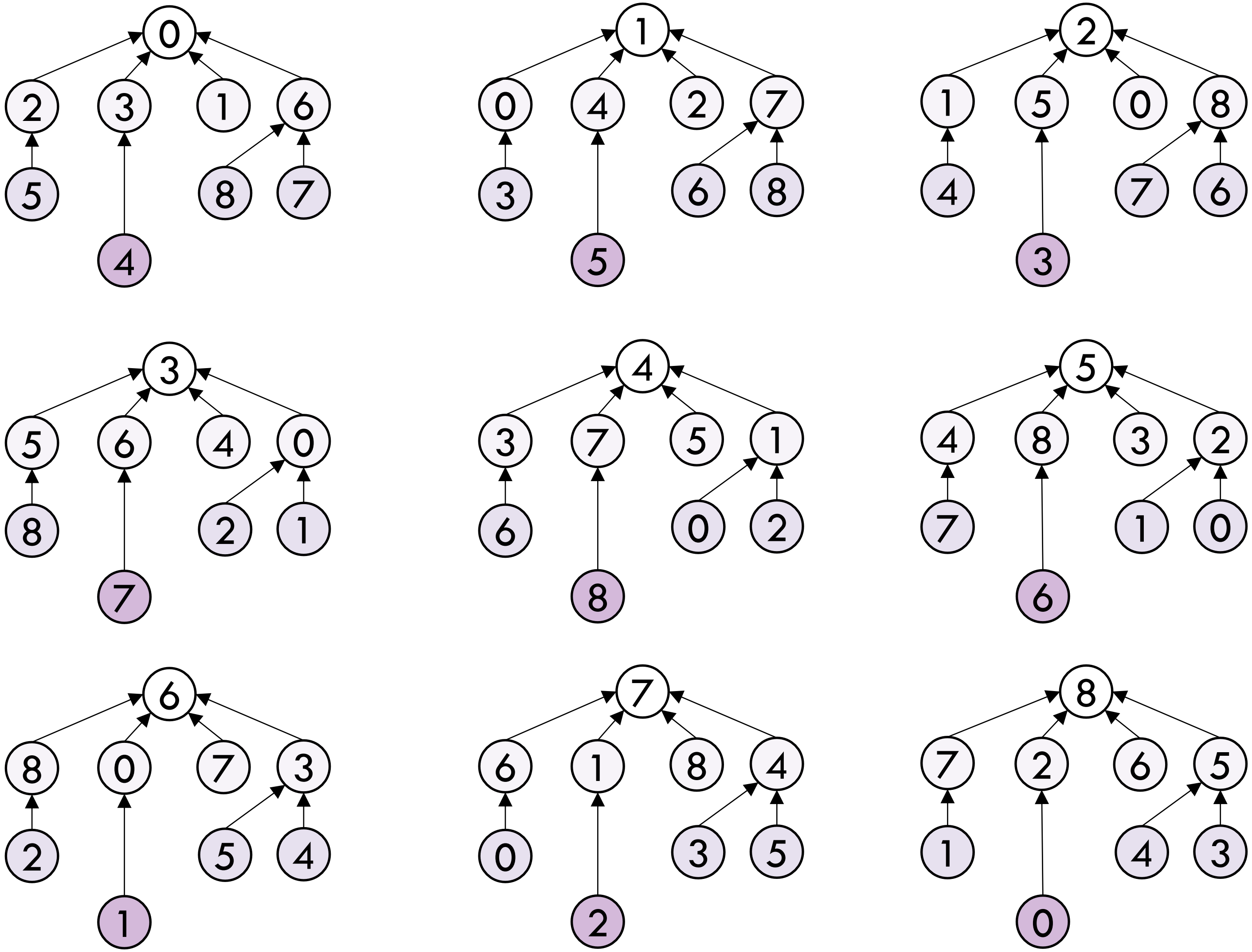
Multitree Reduce-Scatter Phase

X–, Y– Links



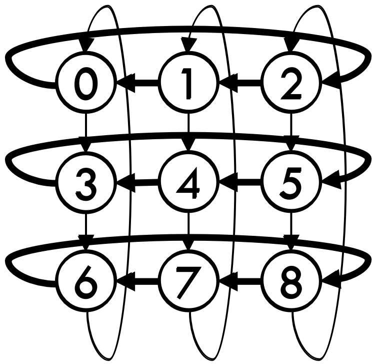


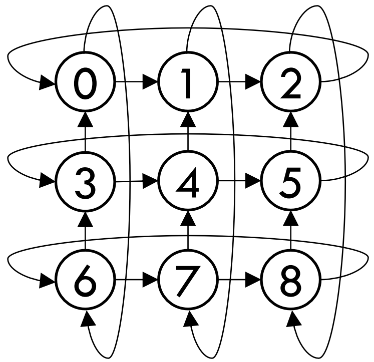
X+, Y+ Links



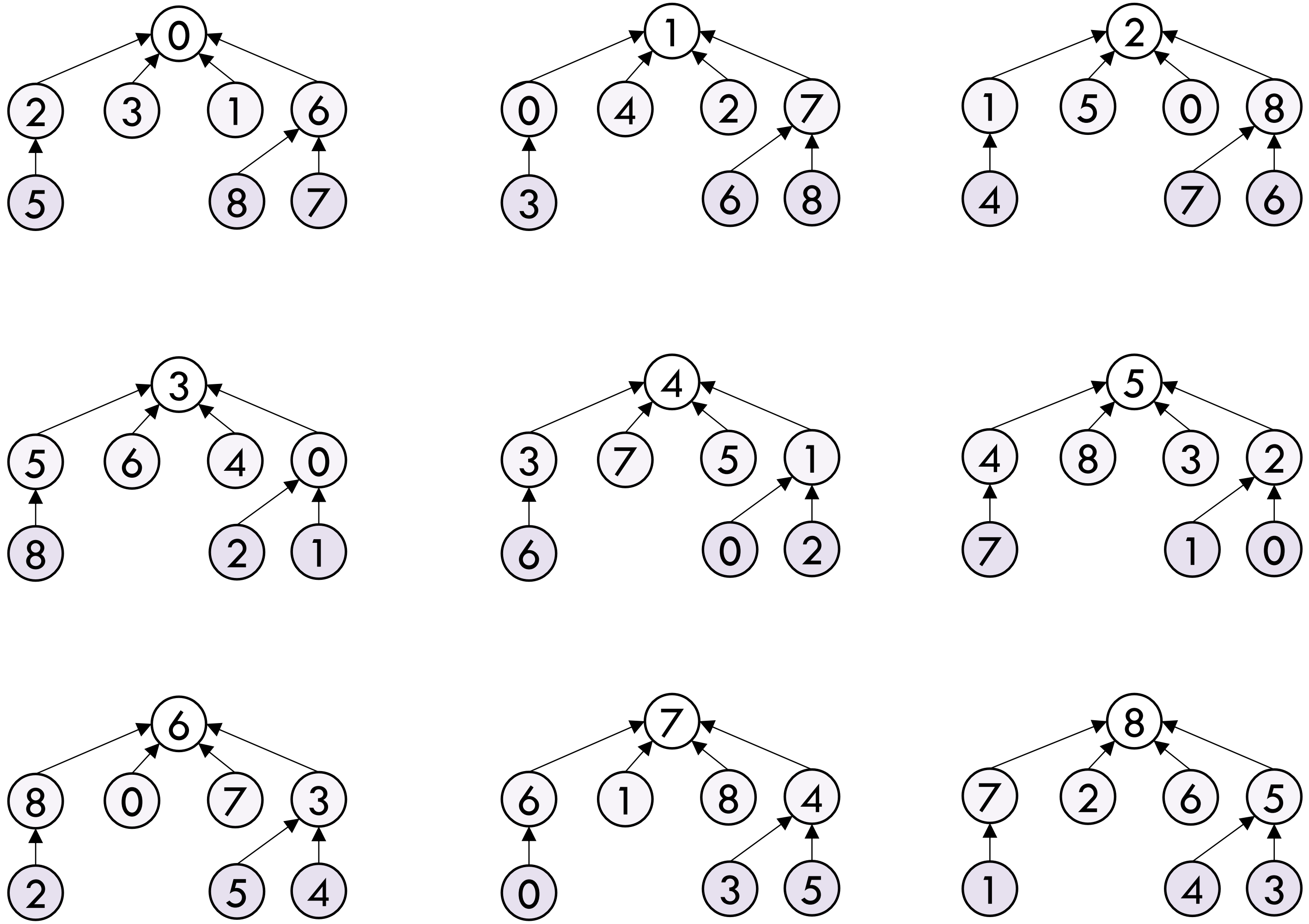
Multitree Reduce-Scatter Phase

X–, Y– Links



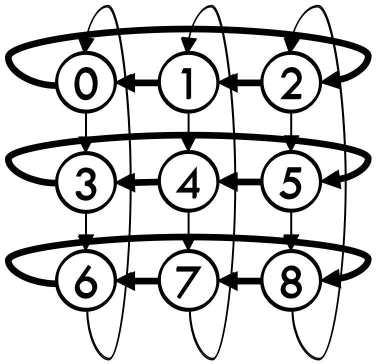


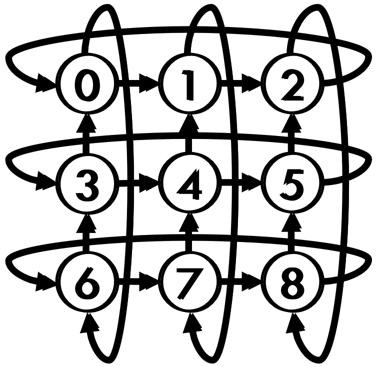
X+, Y+ Links



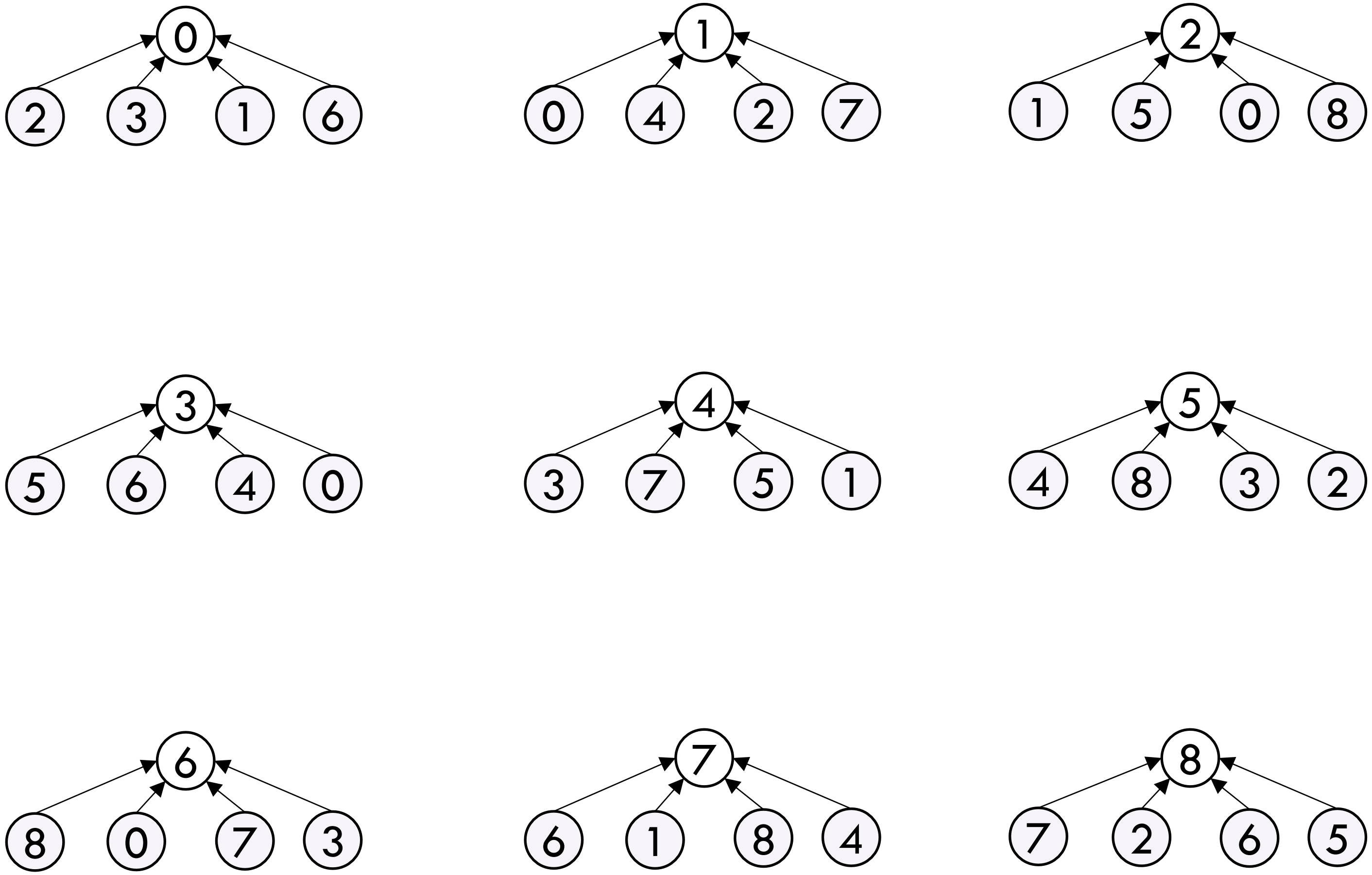
Multitree Reduce-Scatter Phase

X−, Y− Links



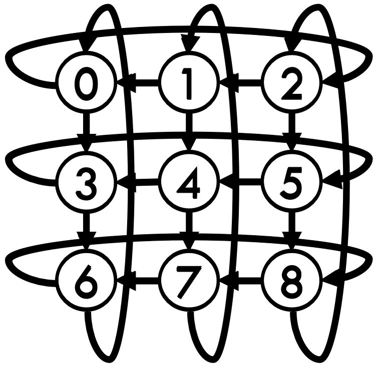


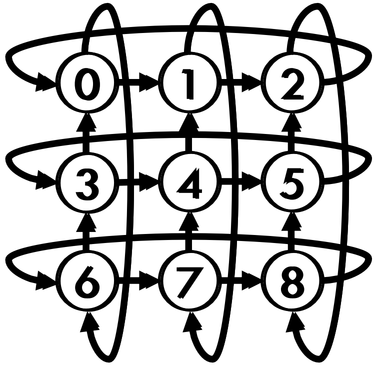
X+, Y+ Links



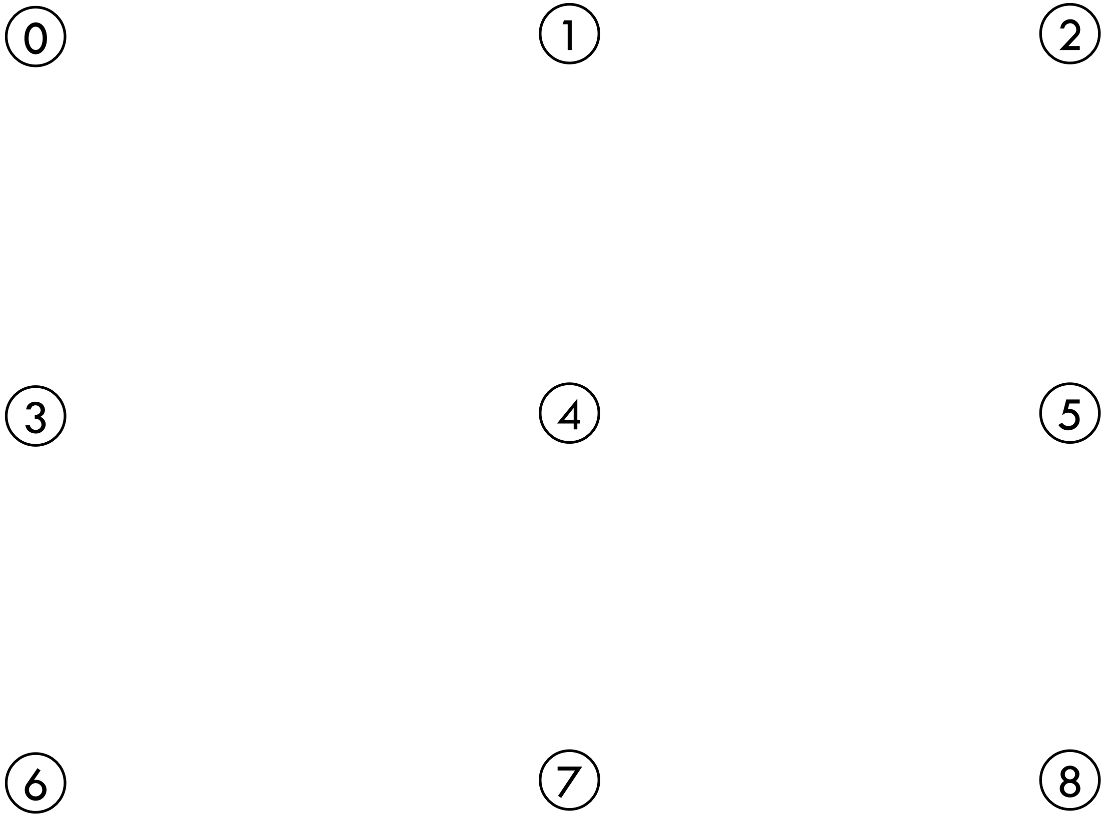
Multitree Reduce-Scatter Phase

X–, Y– Links



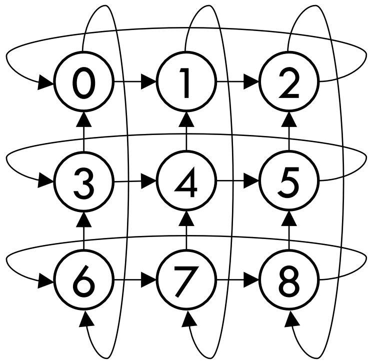


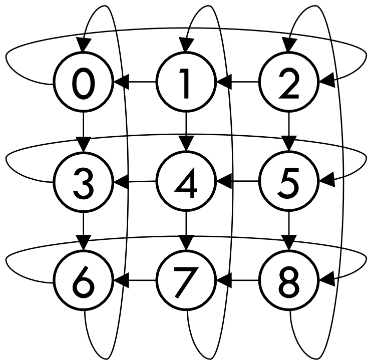
X+, Y+ Links



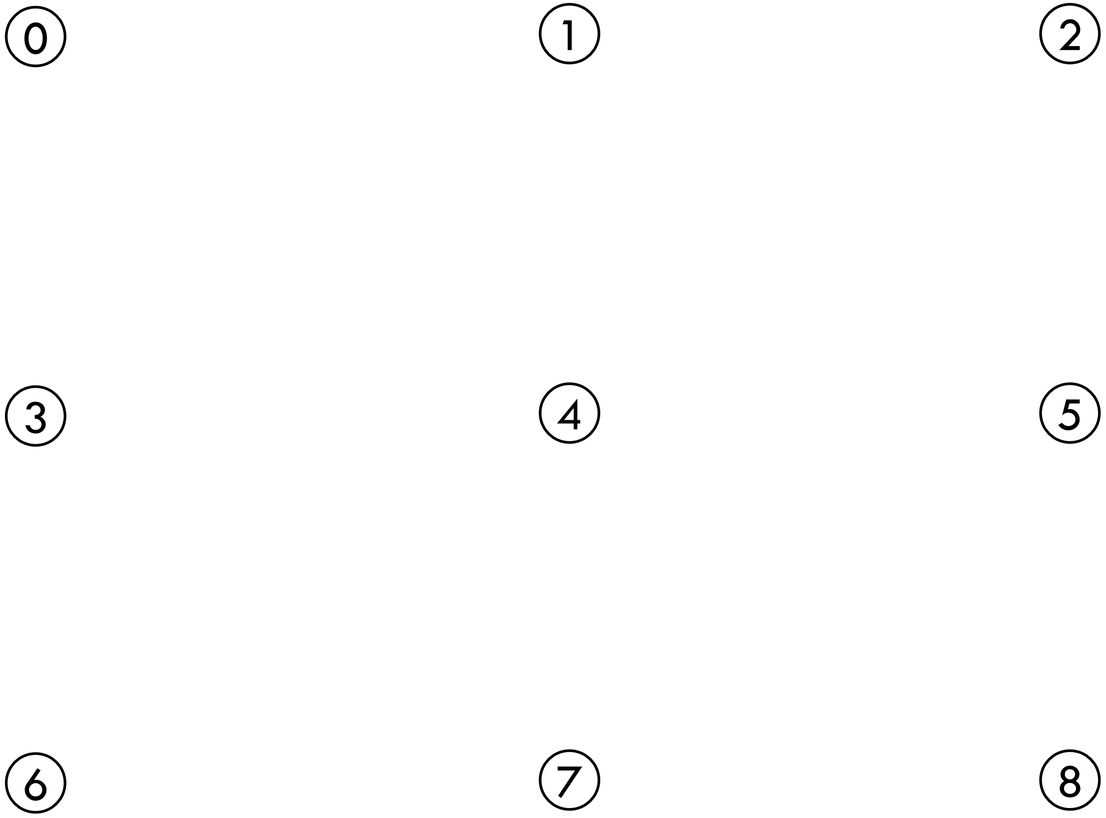
Multitree All-Gather Phase

X+, Y+ Links

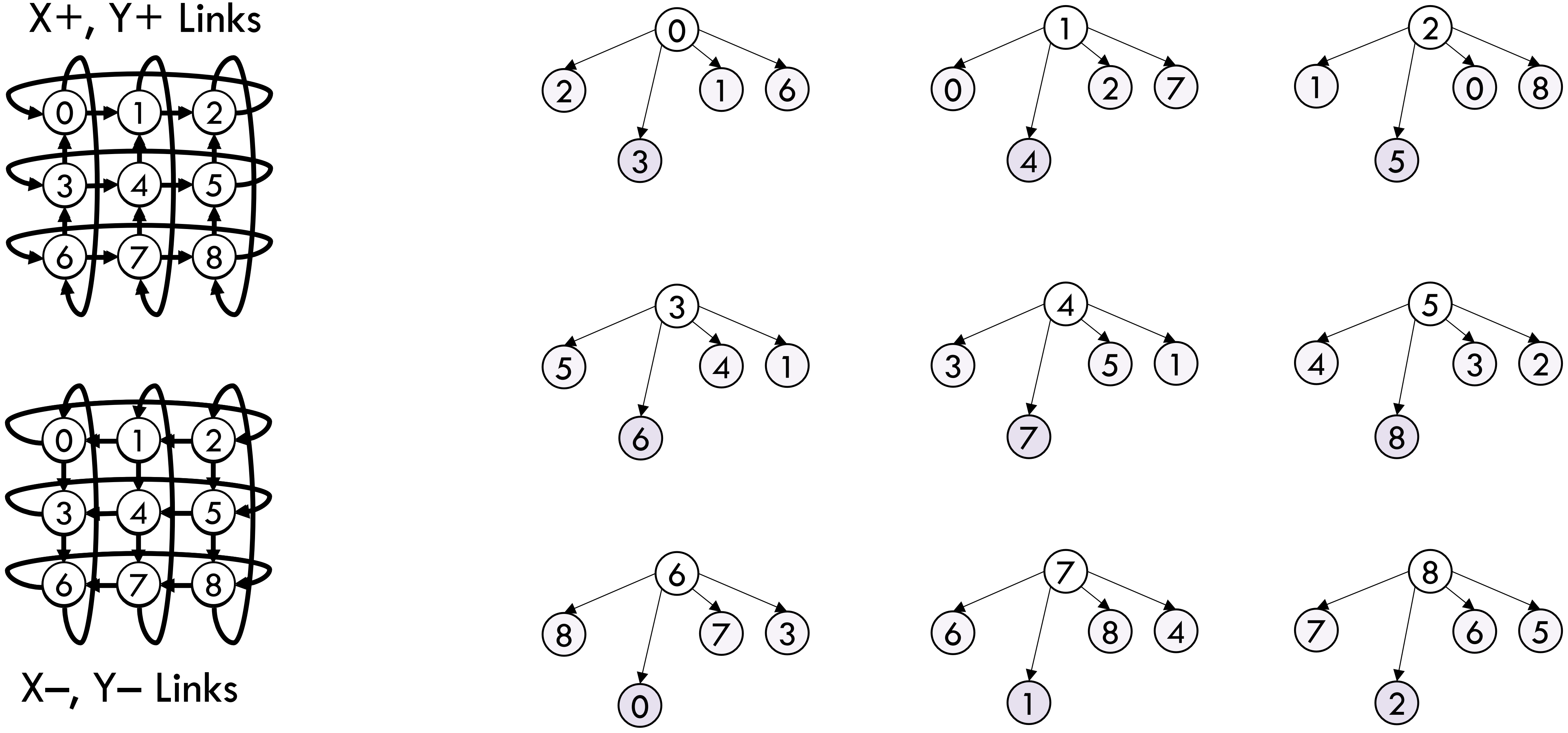




X–, Y– Links

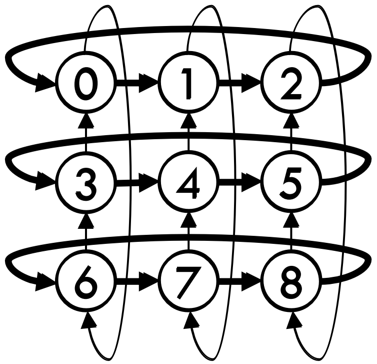


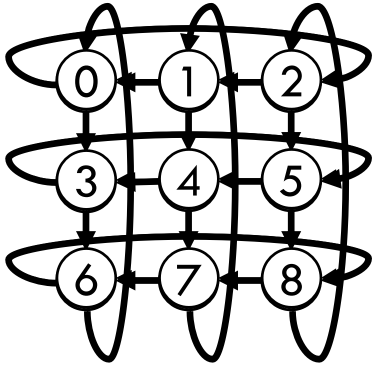
Multitree All-Gather Phase



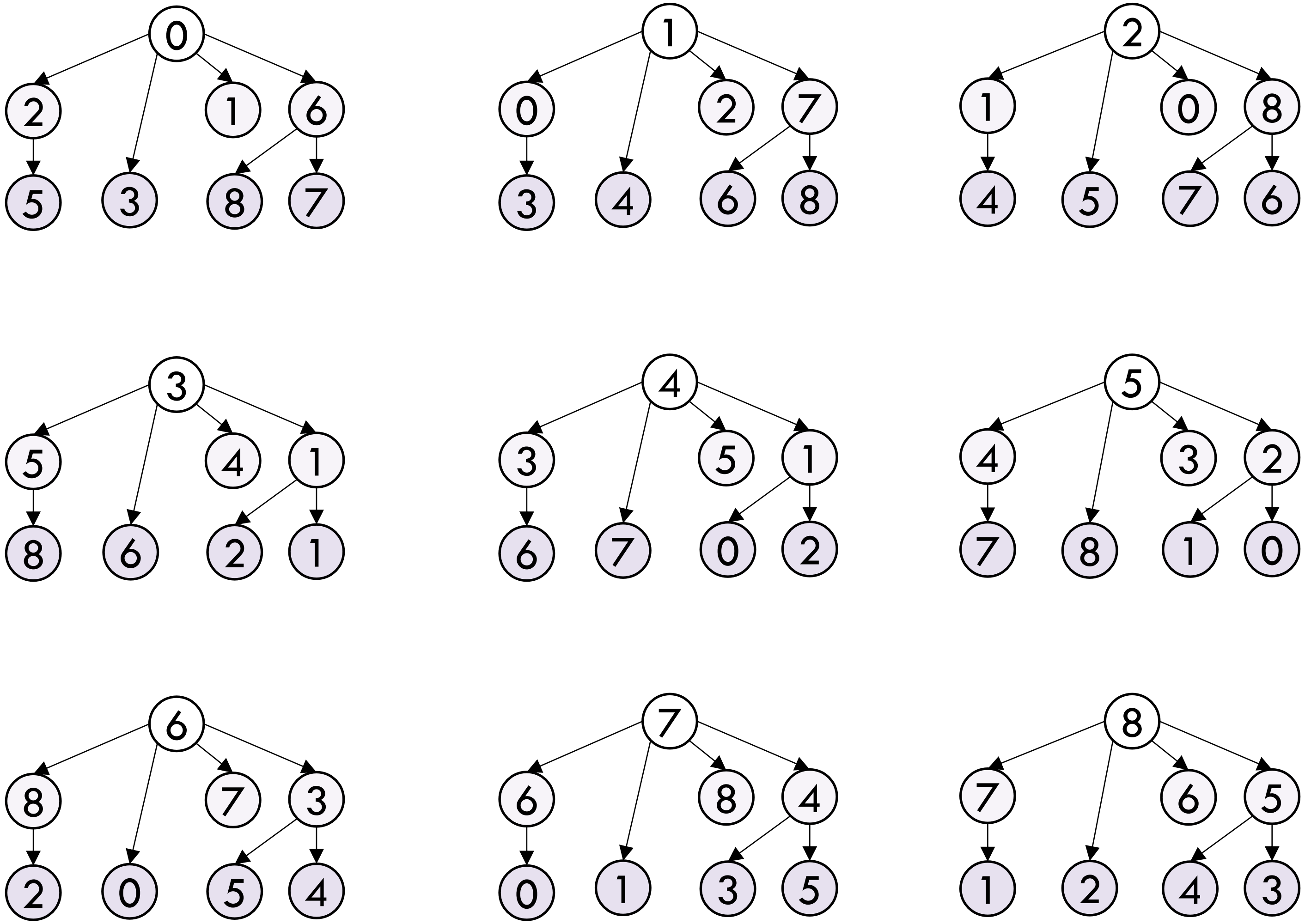
Multitree All-Gather Phase

X+, Y+ Links



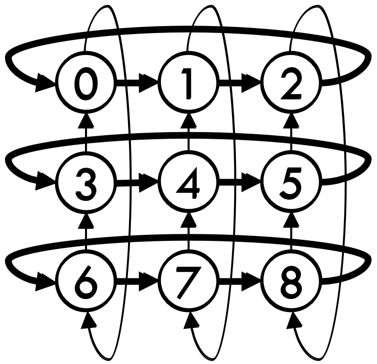


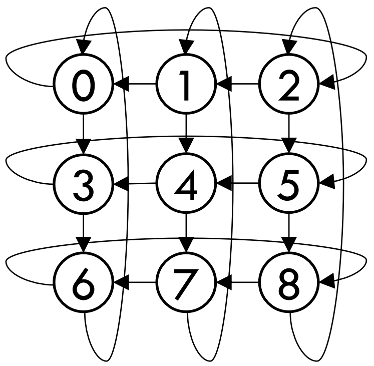
X–, Y– Links



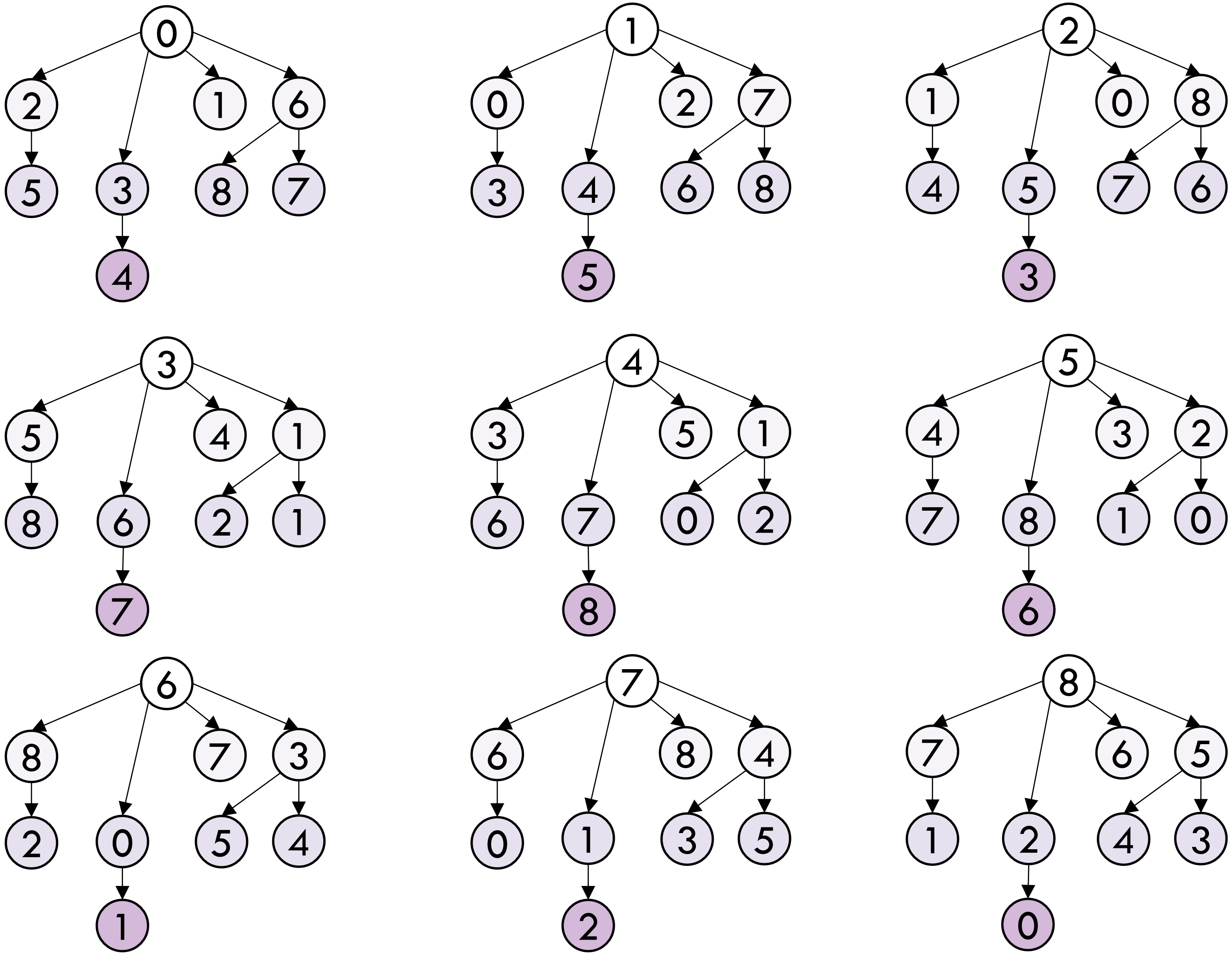
Multitree All-Gather Phase

X+, Y+ Links





X–, Y– Links



Hardware-based All-Reduce Scheduling

- Message Command (Instruction)

Op	FlowID	Parent	Children	Step	Start Addr	Size

  - Stored in an all-reduce schedule table entry
  - Op: `Reduce, Gather, NOP`
  - FlowID: the ID of the reduction/broadcast tree

Hardware-based All-Reduce Scheduling

- Message Command (Instruction)

Op	FlowID	Parent	Children	Step	Start Addr	Size

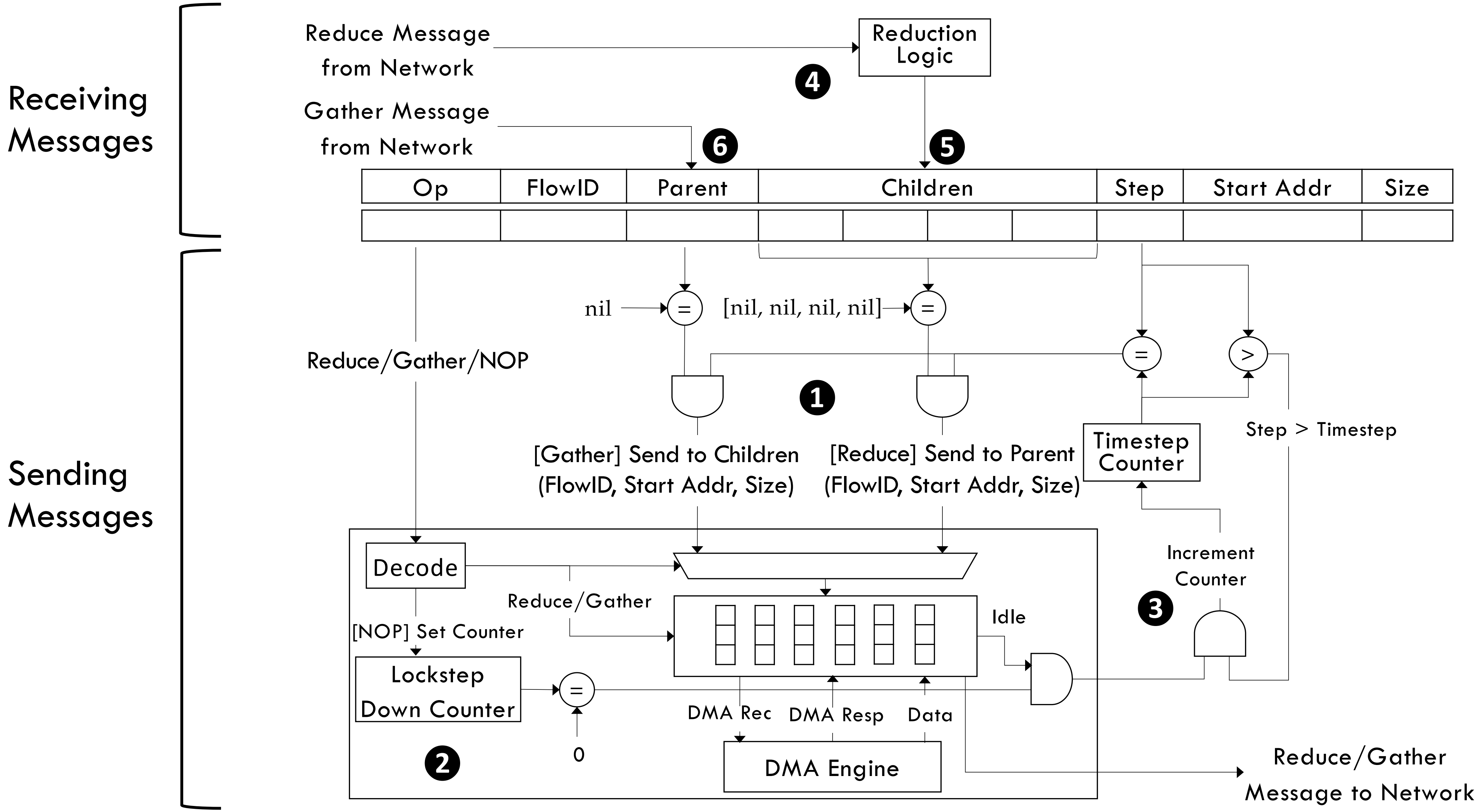
  - Stored in an all-reduce schedule table entry
  - Op: `Reduce, Gather, NOP`
  - FlowID: the ID of the reduction/broadcast tree

Accelerator 0

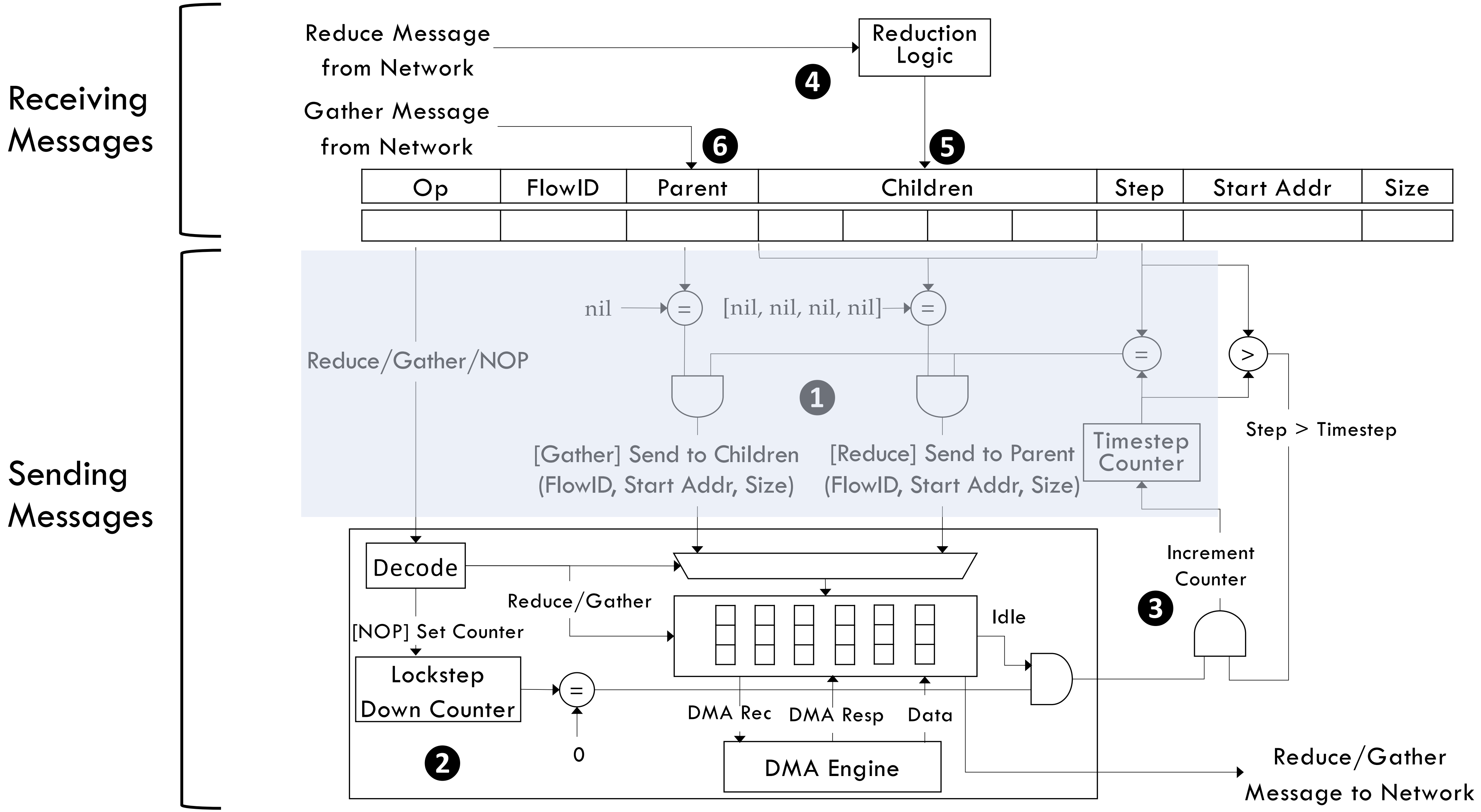
Op	FlowID	Parent	Children				Step
Reduce	8	2	nil	nil	nil	nil	0
Reduce	4	1	nil	nil	nil	nil	1
Reduce	5	2	nil	nil	nil	nil	1
Reduce	7	1	2	nil	nil	nil	1
Reduce	1	1	3	nil	nil	nil	2
Reduce	2	2	nil	nil	nil	nil	2
Reduce	3	3	1	2	nil	nil	2
Reduce	6	6	1	nil	nil	nil	2
Reduce	0	nil	1	2	3	6	2

Op	FlowID	Parent	Children				Step
Gather	0	nil	1	2	3	6	3
Gather	1	1	3	nil	nil	nil	4
Gather	2	2	nil	nil	nil	nil	4
Gather	4	1	nil	nil	nil	nil	5
Gather	5	2	nil	nil	nil	nil	5
Gather	6	6	1	nil	nil	nil	5
Gather	7	6	nil	nil	nil	nil	5
Gather	8	2	nil	nil	nil	nil	5

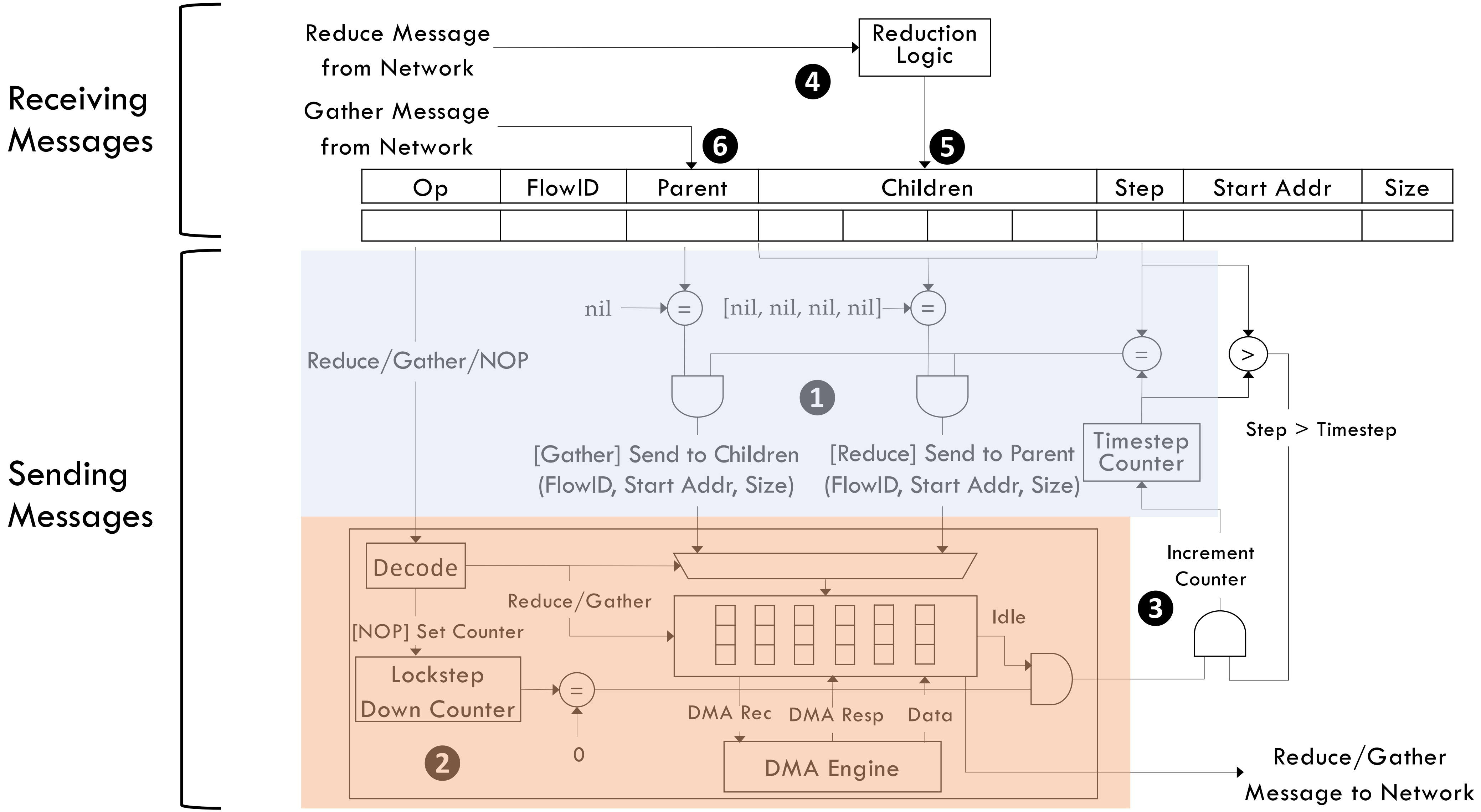
All-Reduce Schedule Control and Datapath



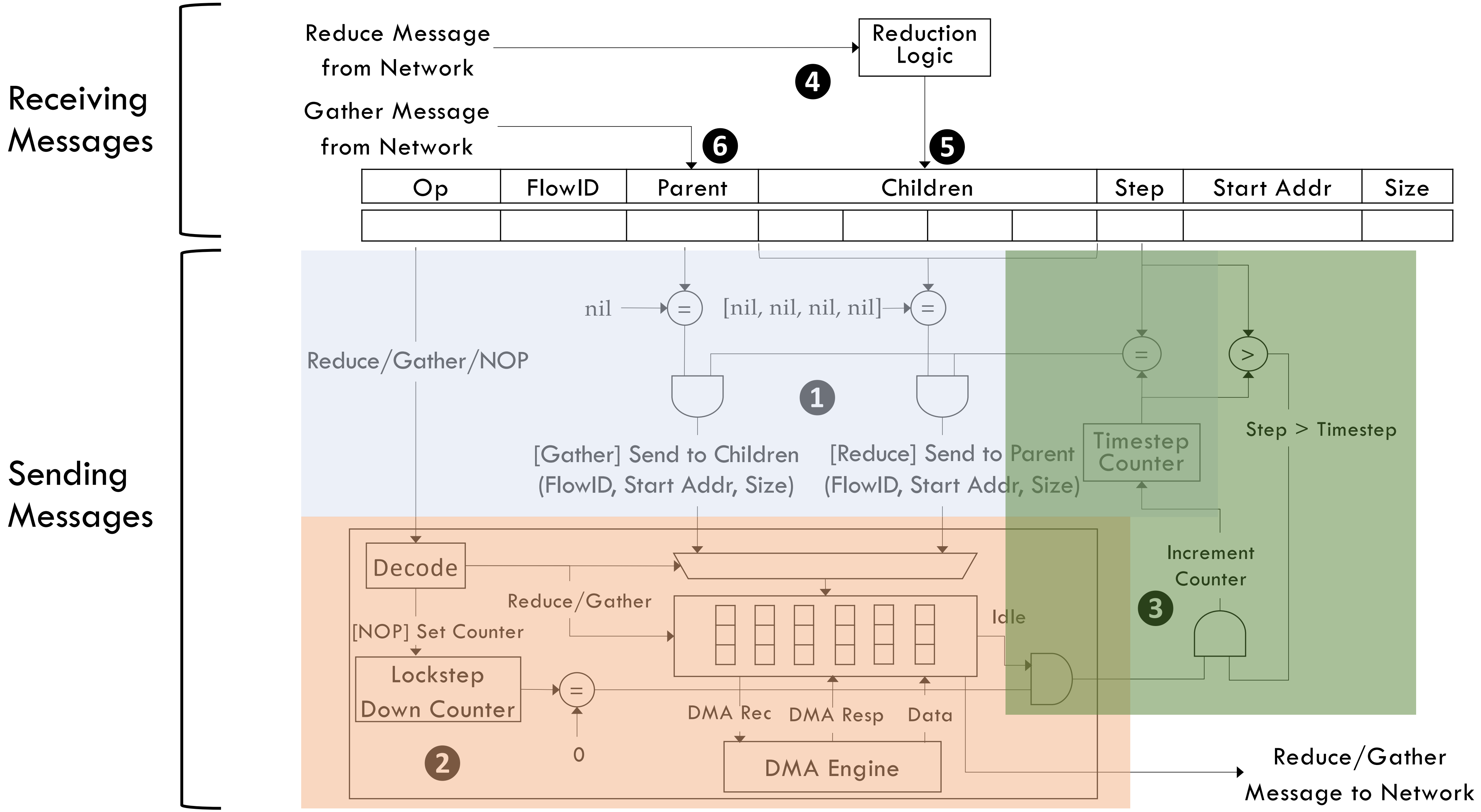
All-Reduce Schedule Control and Datapath



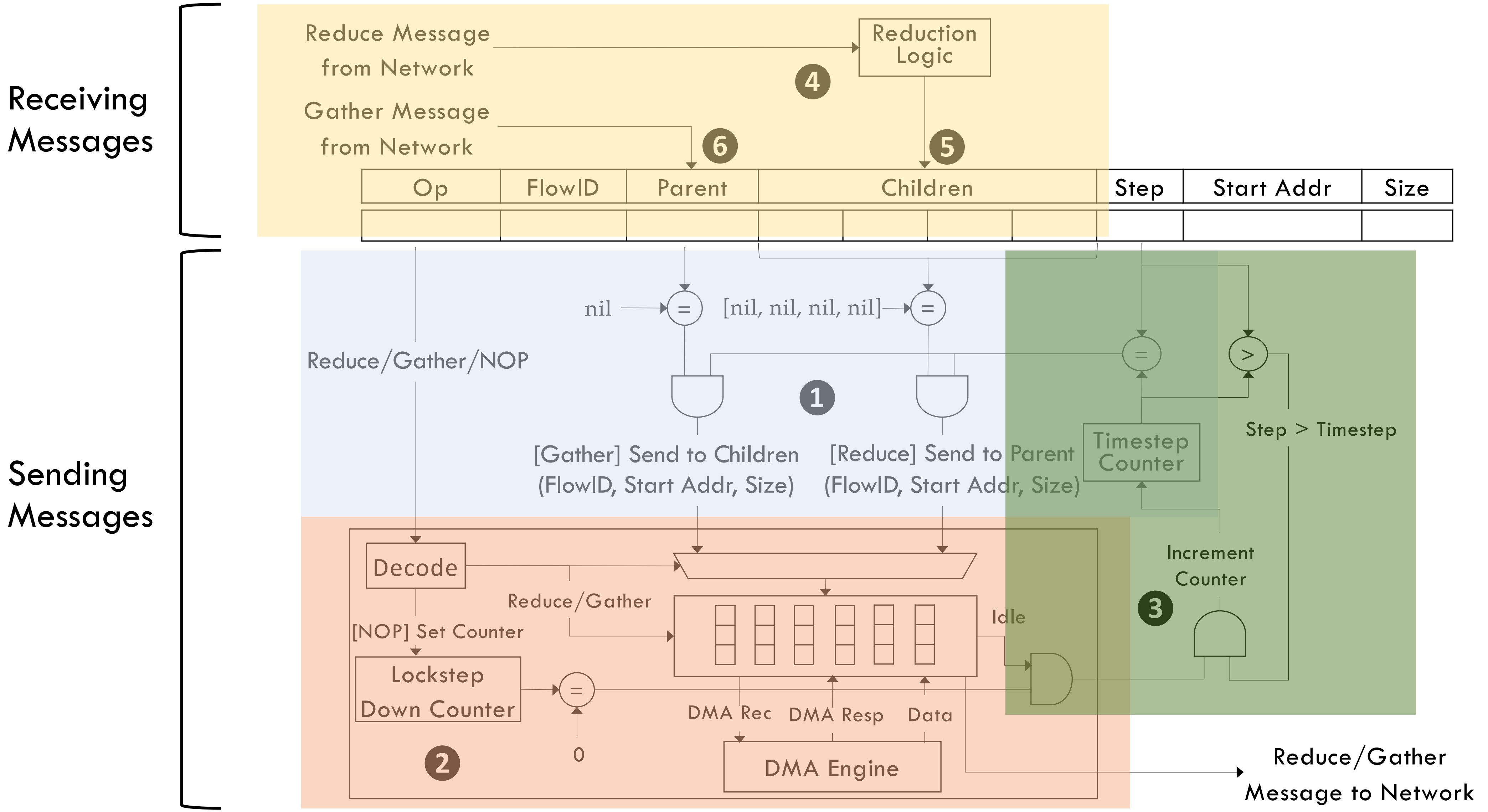
All-Reduce Schedule Control and Datapath



All-Reduce Schedule Control and Datapath

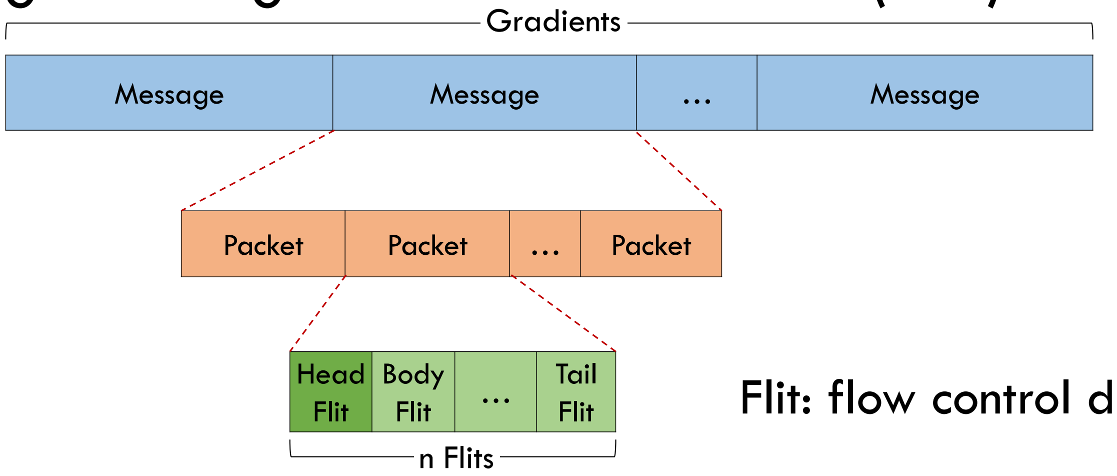


All-Reduce Schedule Control and Datapath



Big Message Flow Control

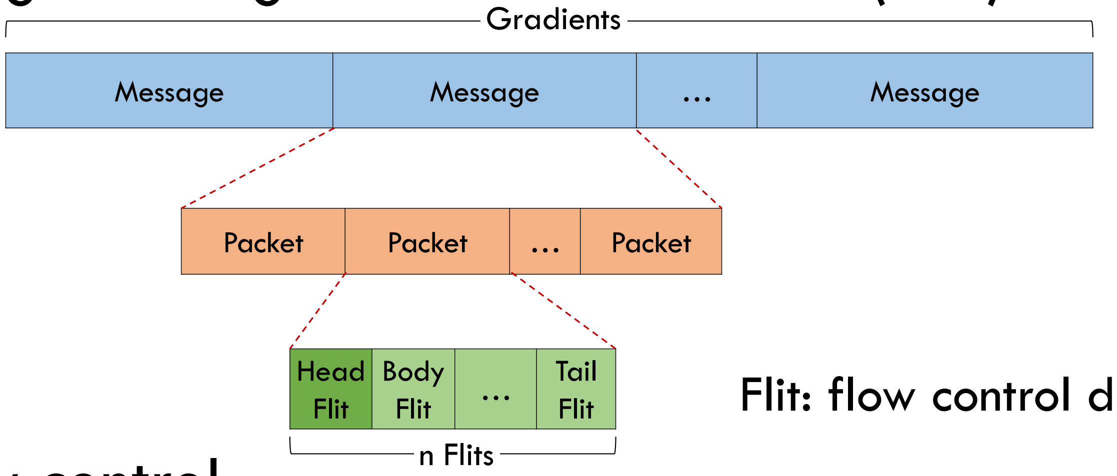
- Big gradient exchange has high head flit overhead (6%)



Flit: flow control digit

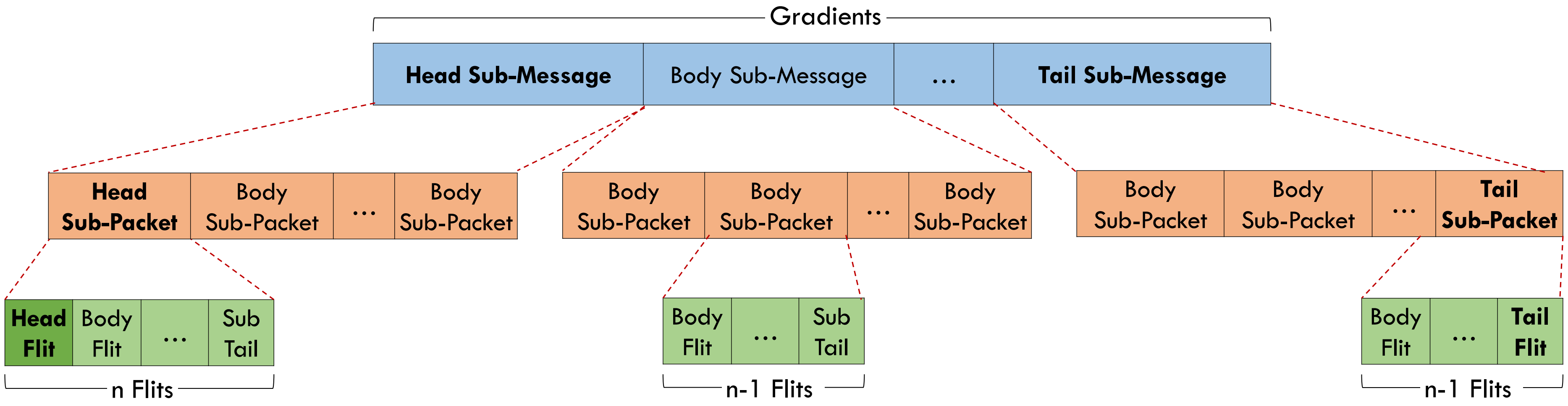
Big Message Flow Control

- Big gradient exchange has high head flit overhead (6%)

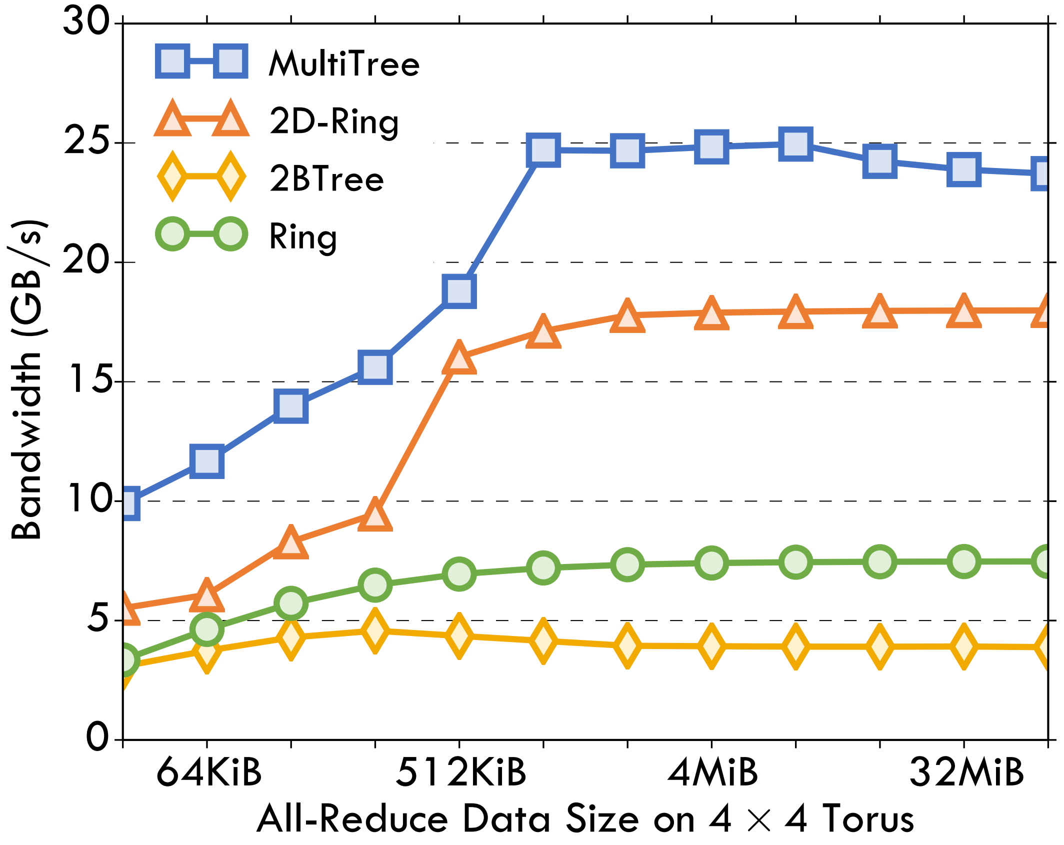


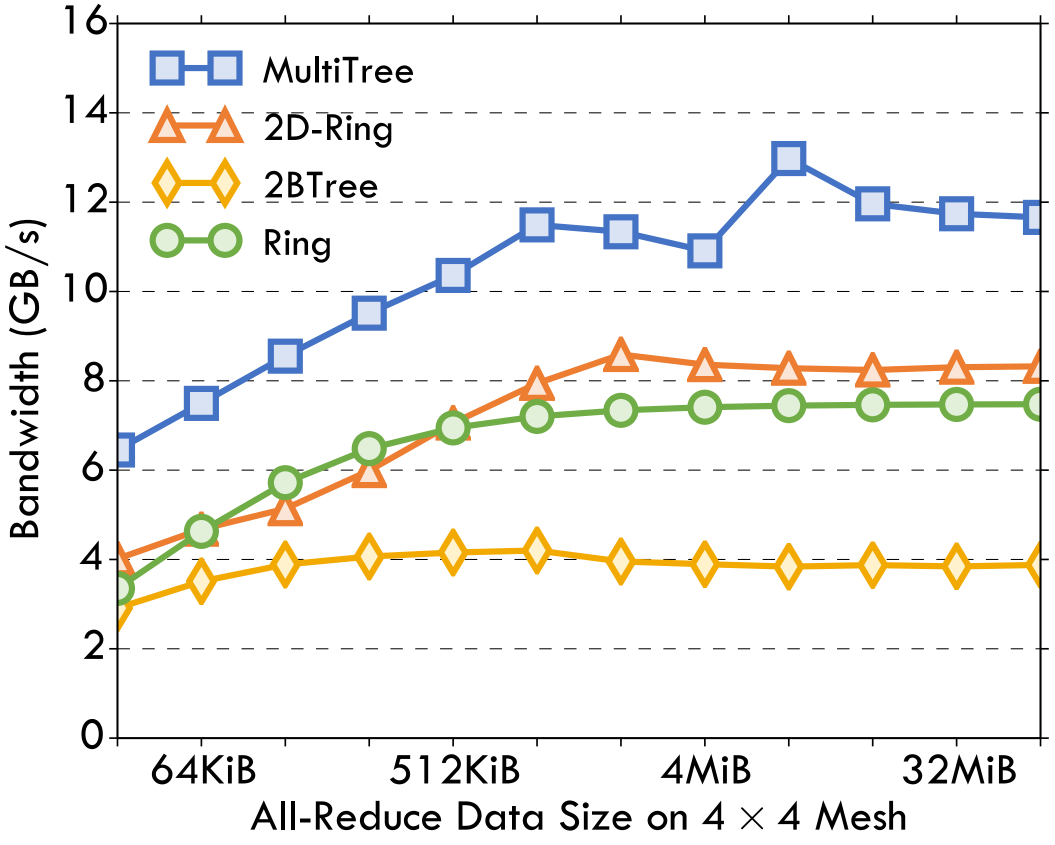
Flit: flow control digit

- Message-based flow control

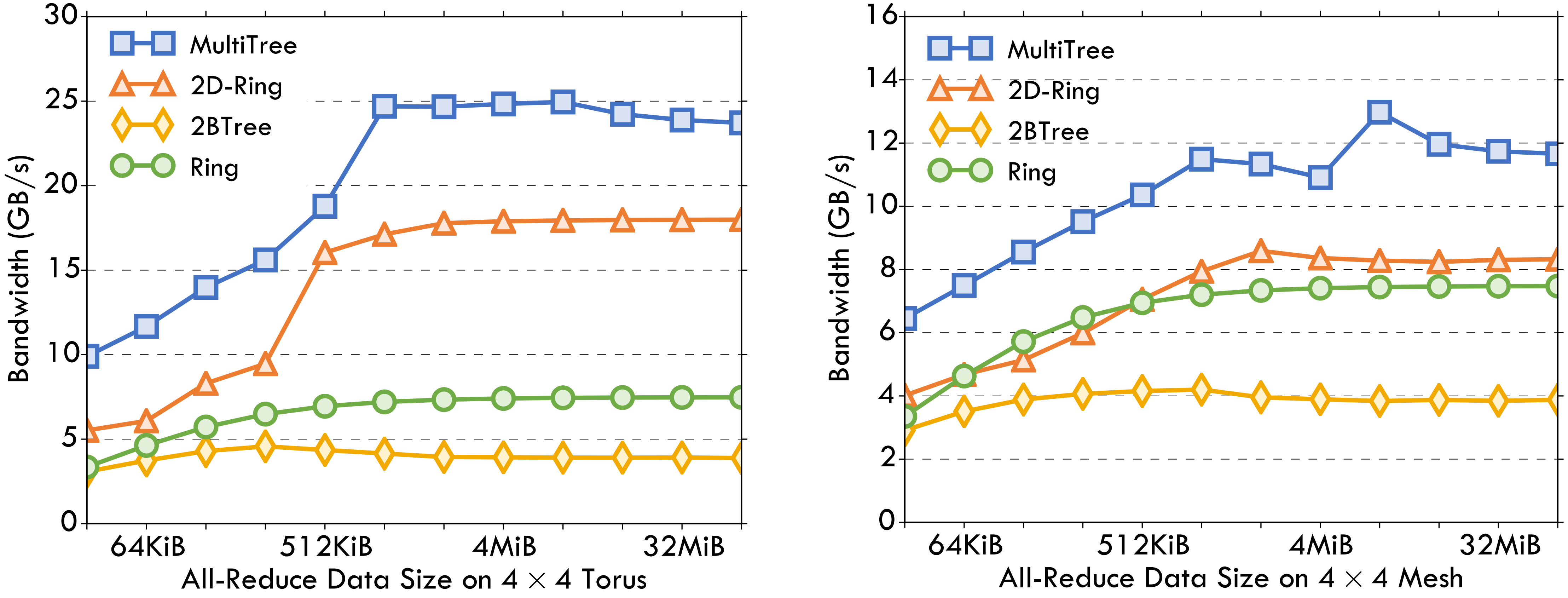


Results – Bandwidth on Directed Networks



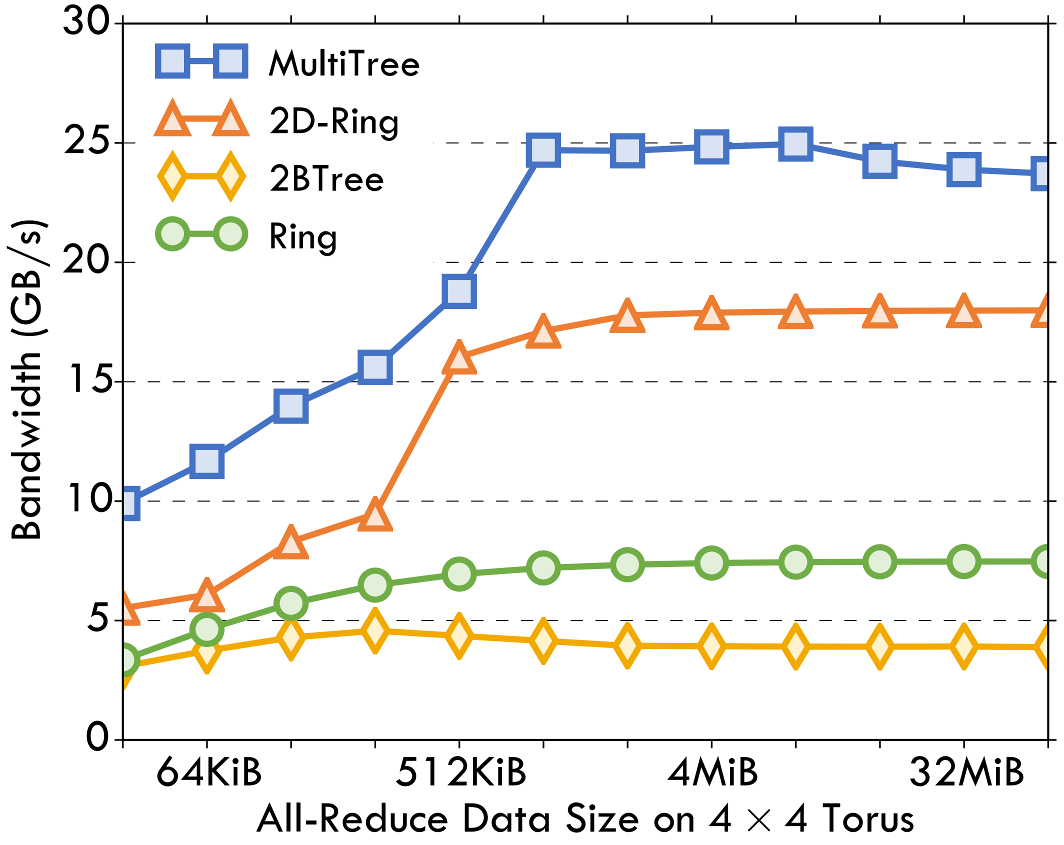


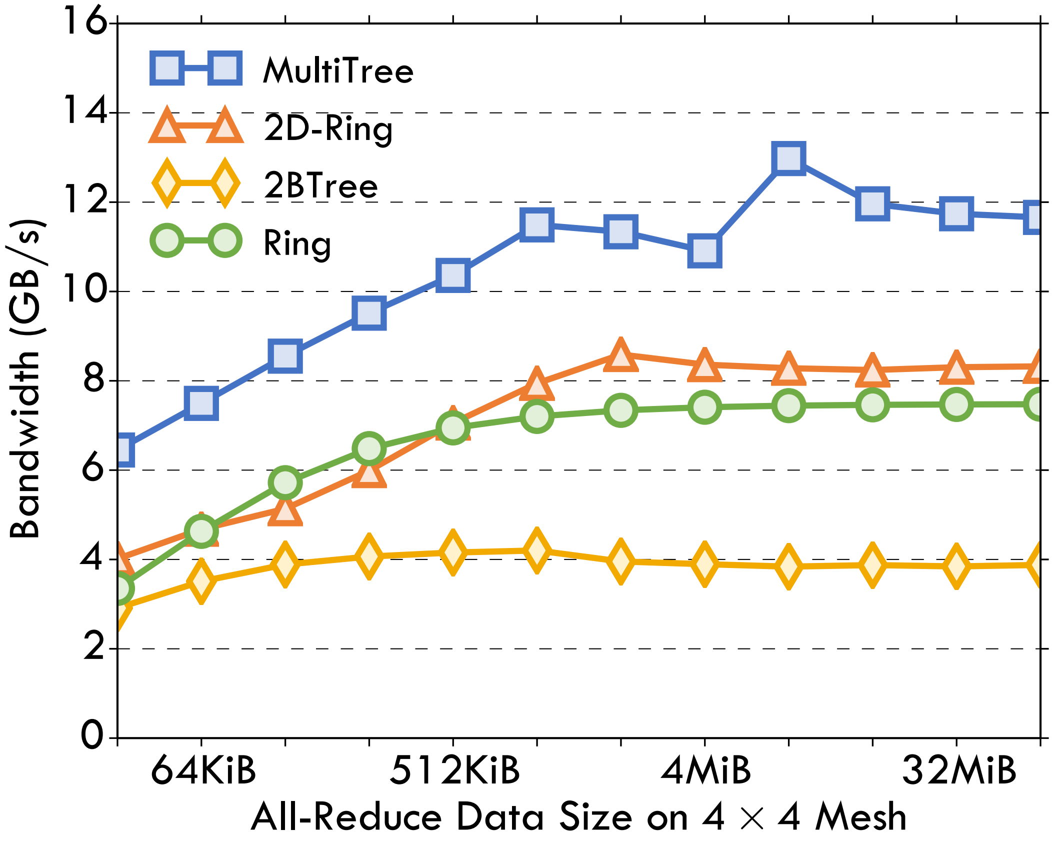
Results – Bandwidth on Directed Networks



- Double binary tree (2BTree) is very unfriendly to Torus and Mesh

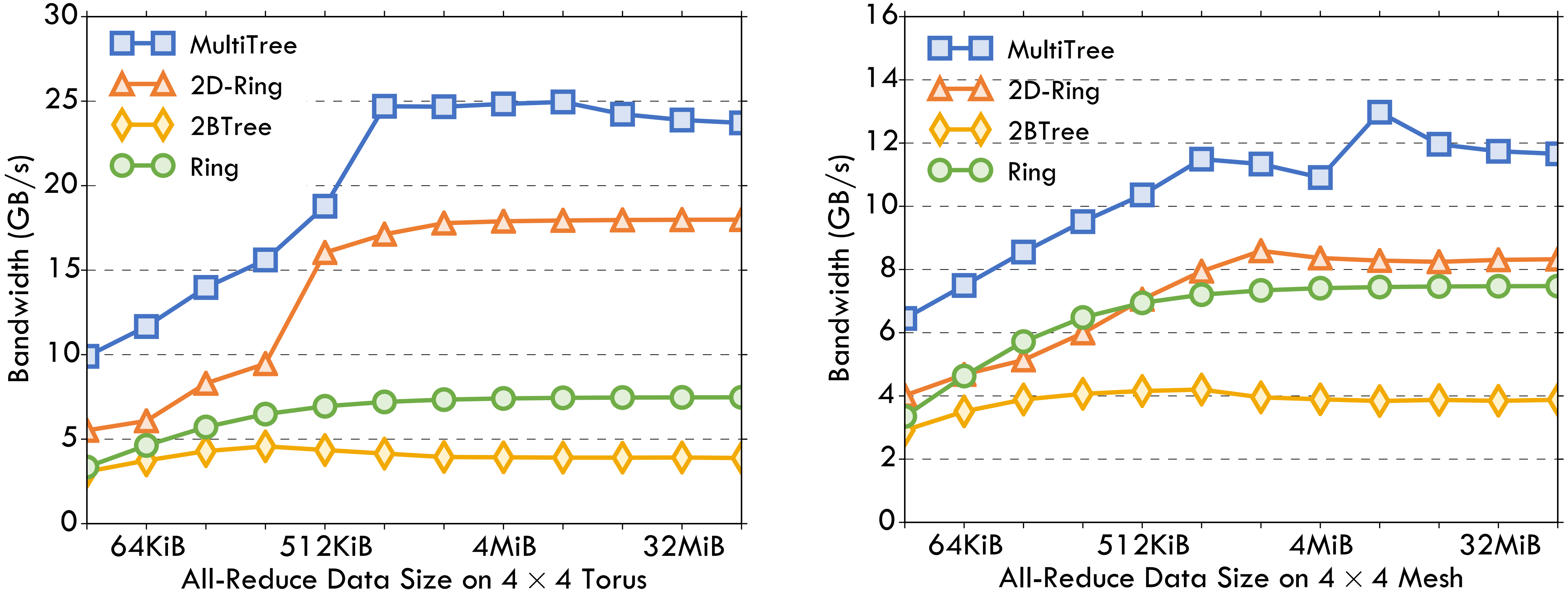
Results – Bandwidth on Directed Networks





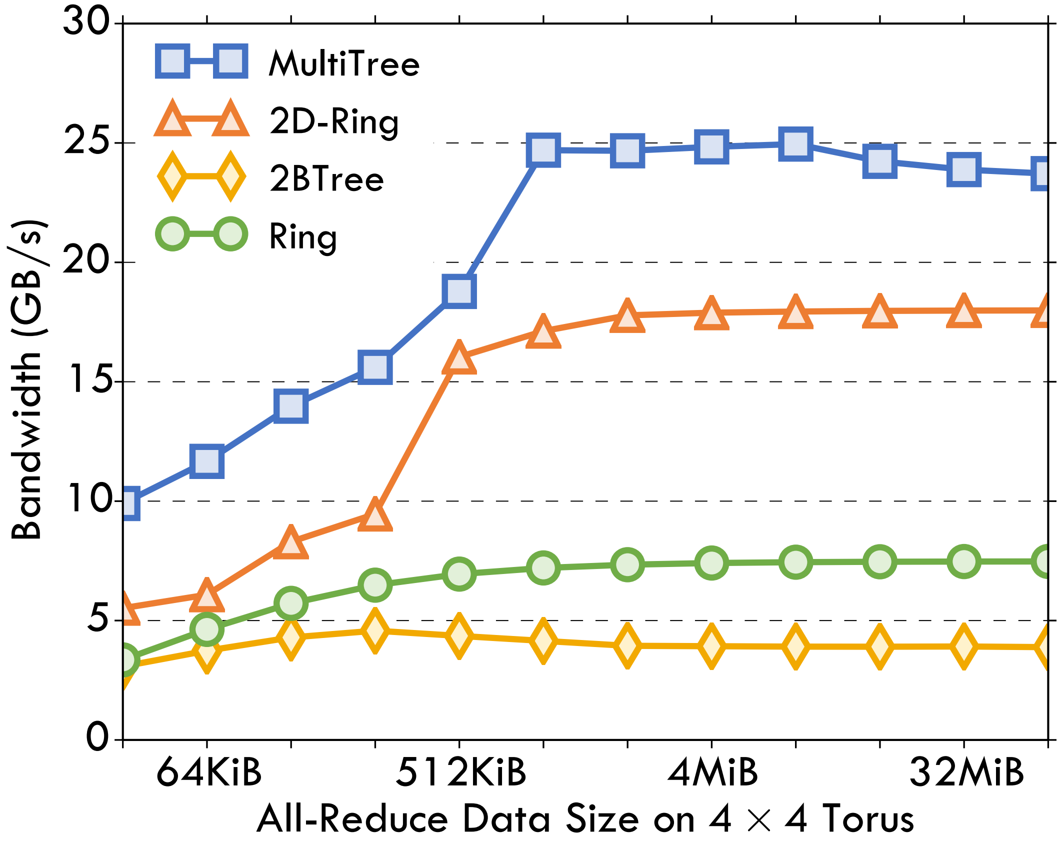
- Double binary tree (2BTree) is very unfriendly to Torus and Mesh
- Ring faces sever link under utilizations

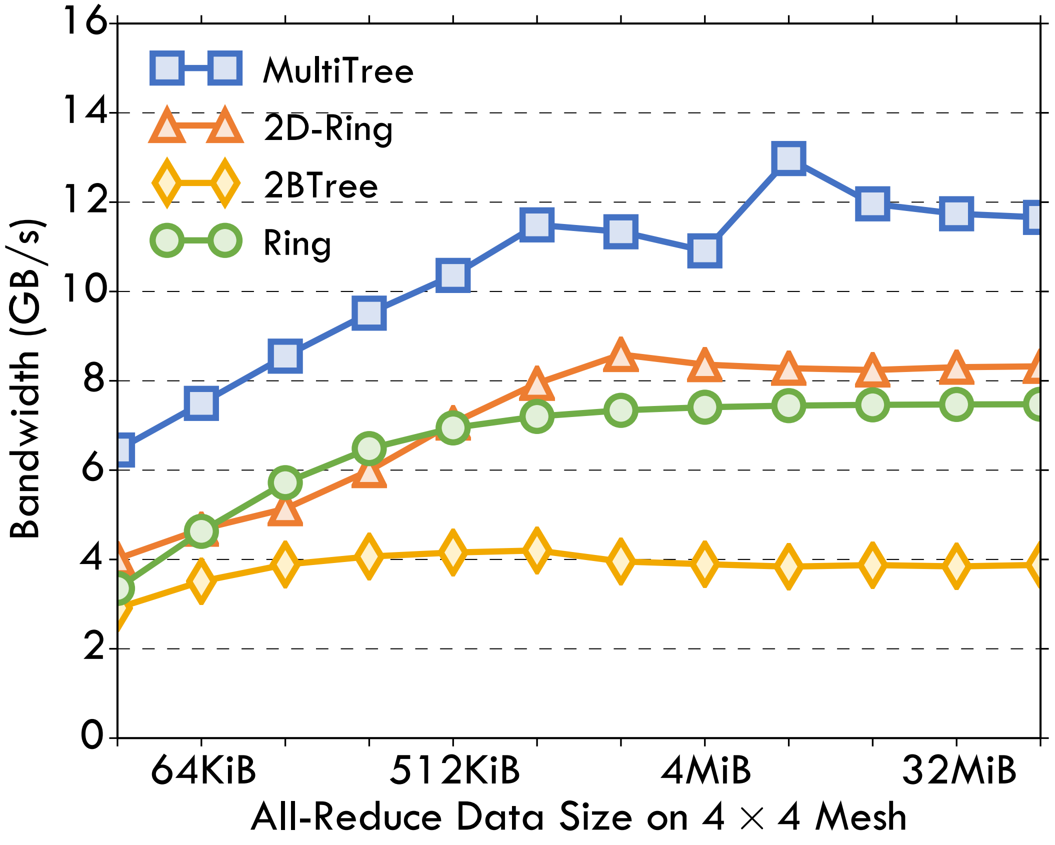
Results – Bandwidth on Directed Networks



- Double binary tree (2BTree) is very unfriendly to Torus and Mesh
- Ring faces sever link under utilizations
- 2D-Ring bandwidth sub-optimal, sending more data (could be twice amount)

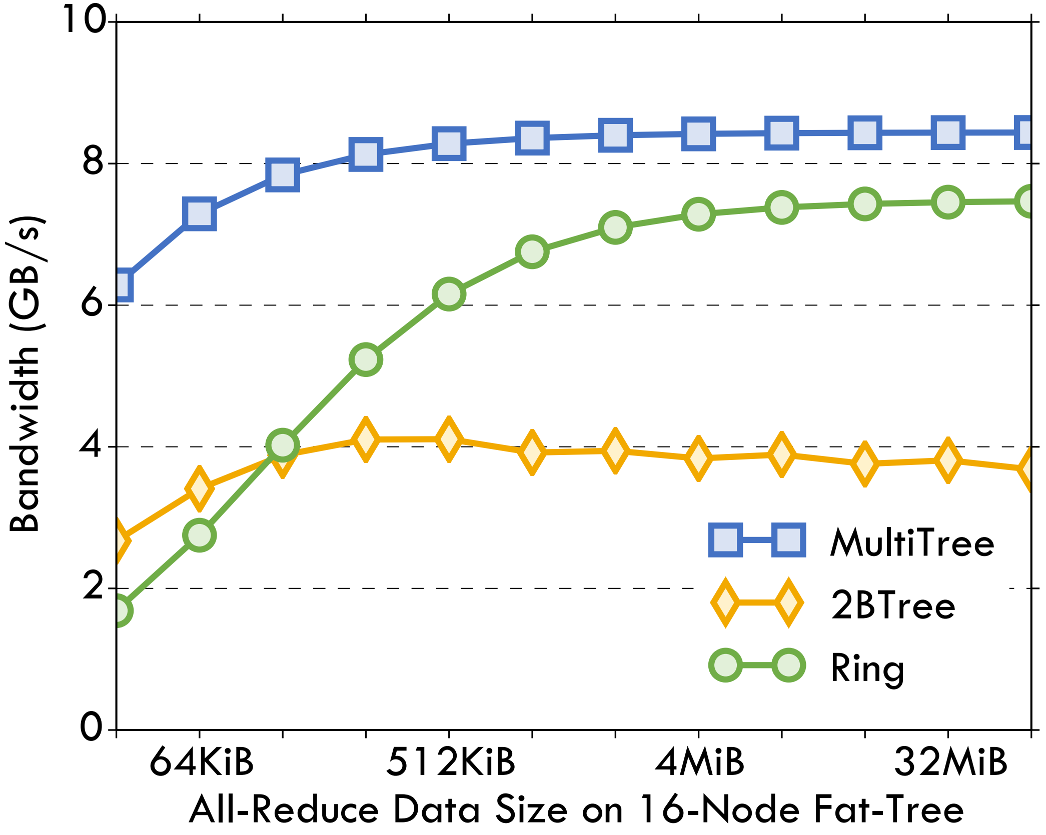
Results – Bandwidth on Directed Networks

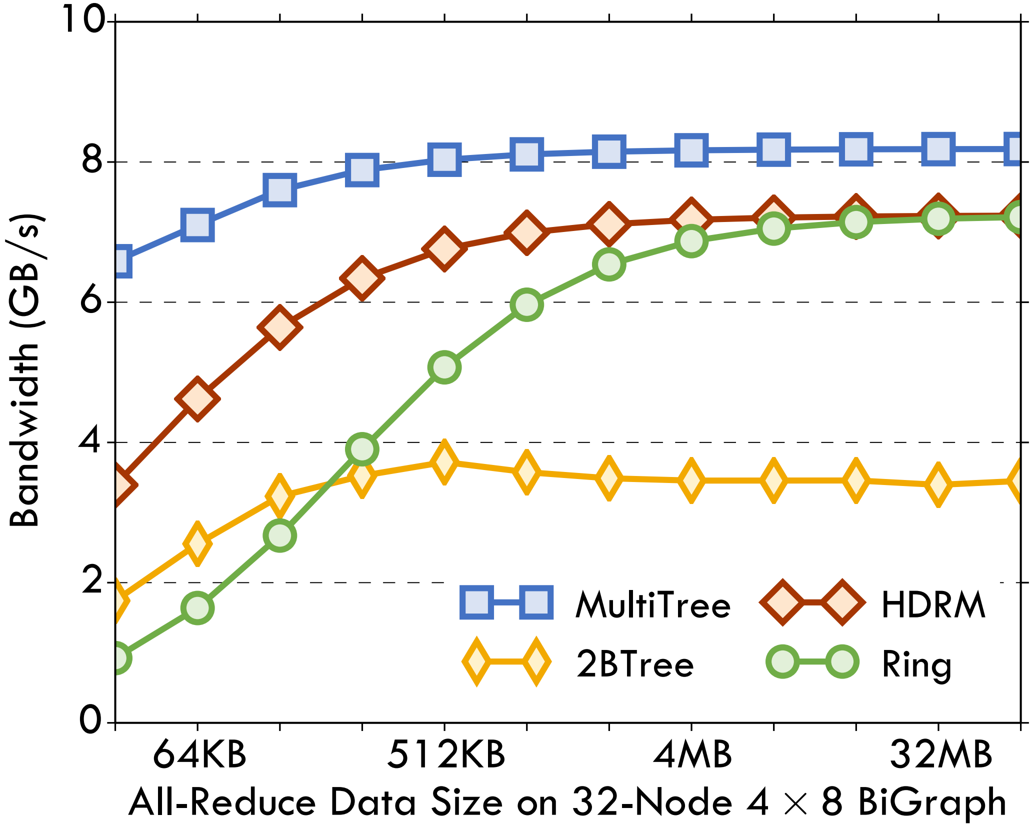




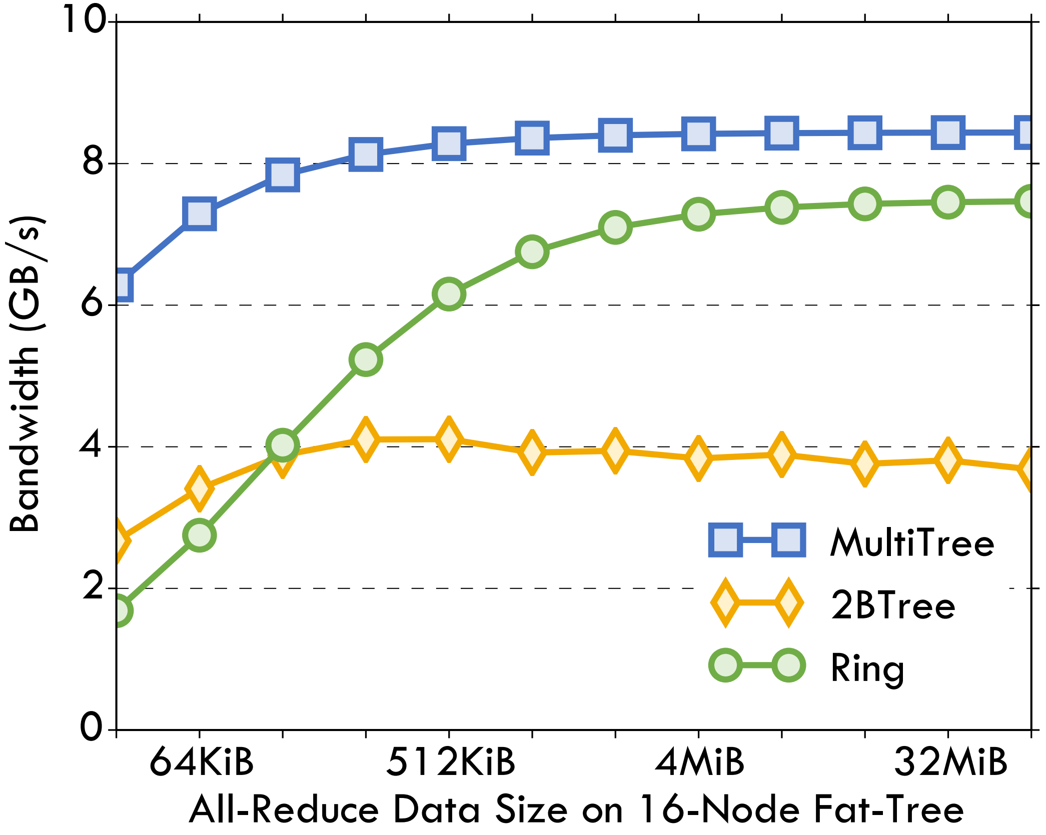
- Double binary tree (2BTree) is very unfriendly to Torus and Mesh
- Ring faces sever link under utilizations
- 2D-Ring bandwidth sub-optimal, sending more data (could be twice amount)
- MultiTree solves all the above problems

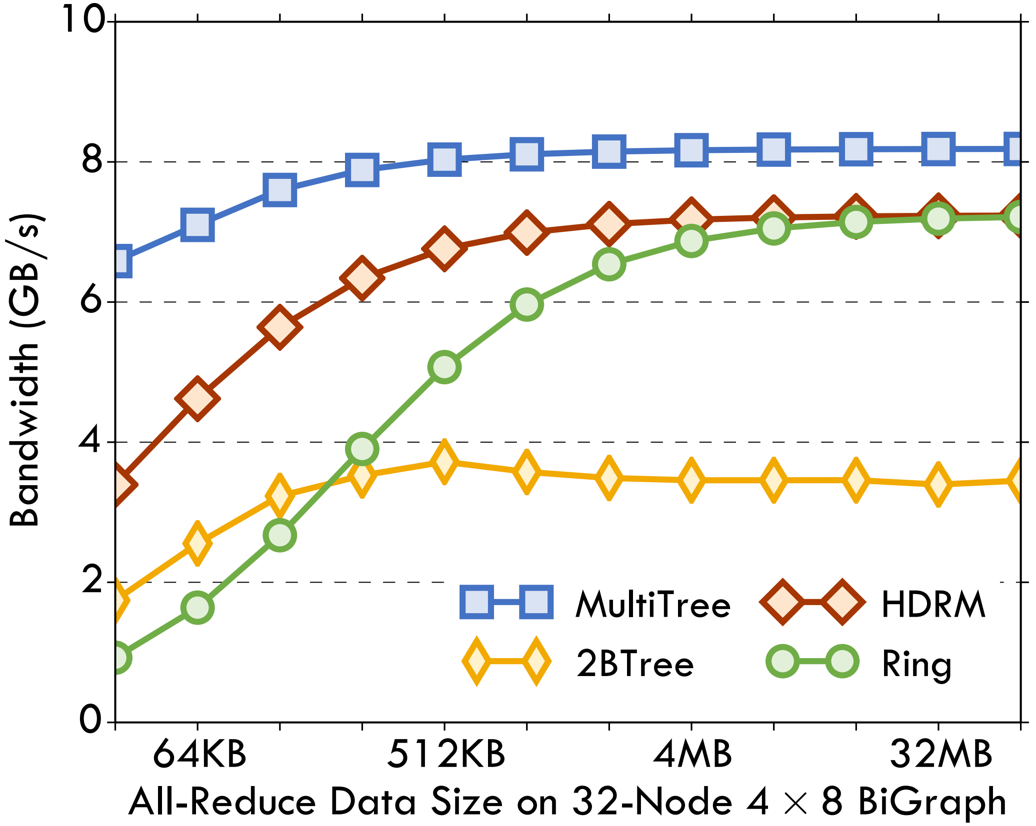
Results – Bandwidth on Switch-based Networks





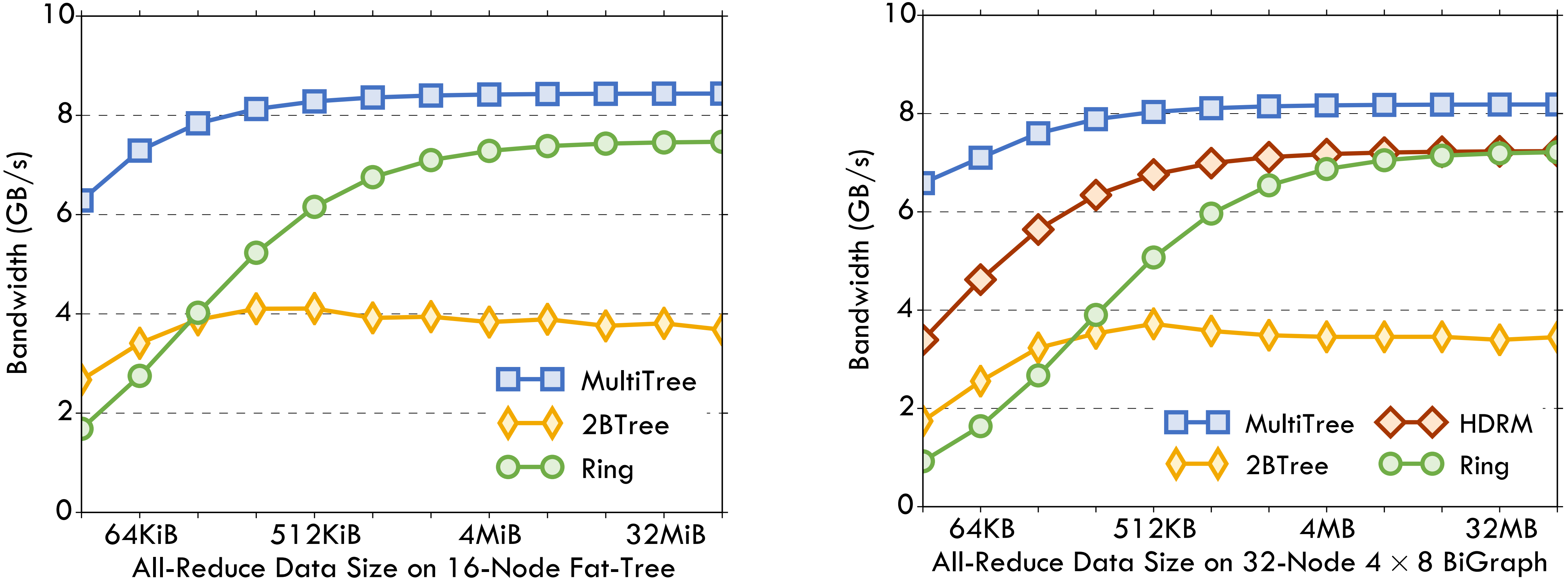
Results – Bandwidth on Switch-based Networks





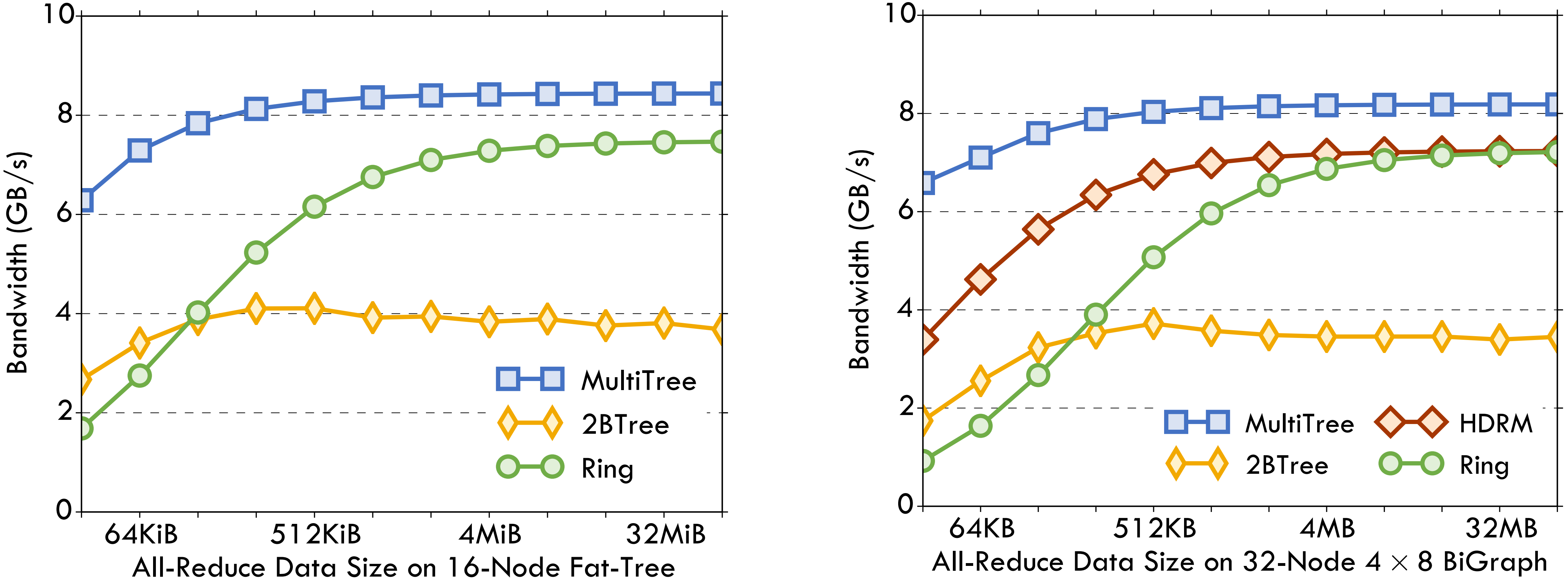
- Ring achieves good bandwidth for large data (long latency for small data)

Results – Bandwidth on Switch-based Networks



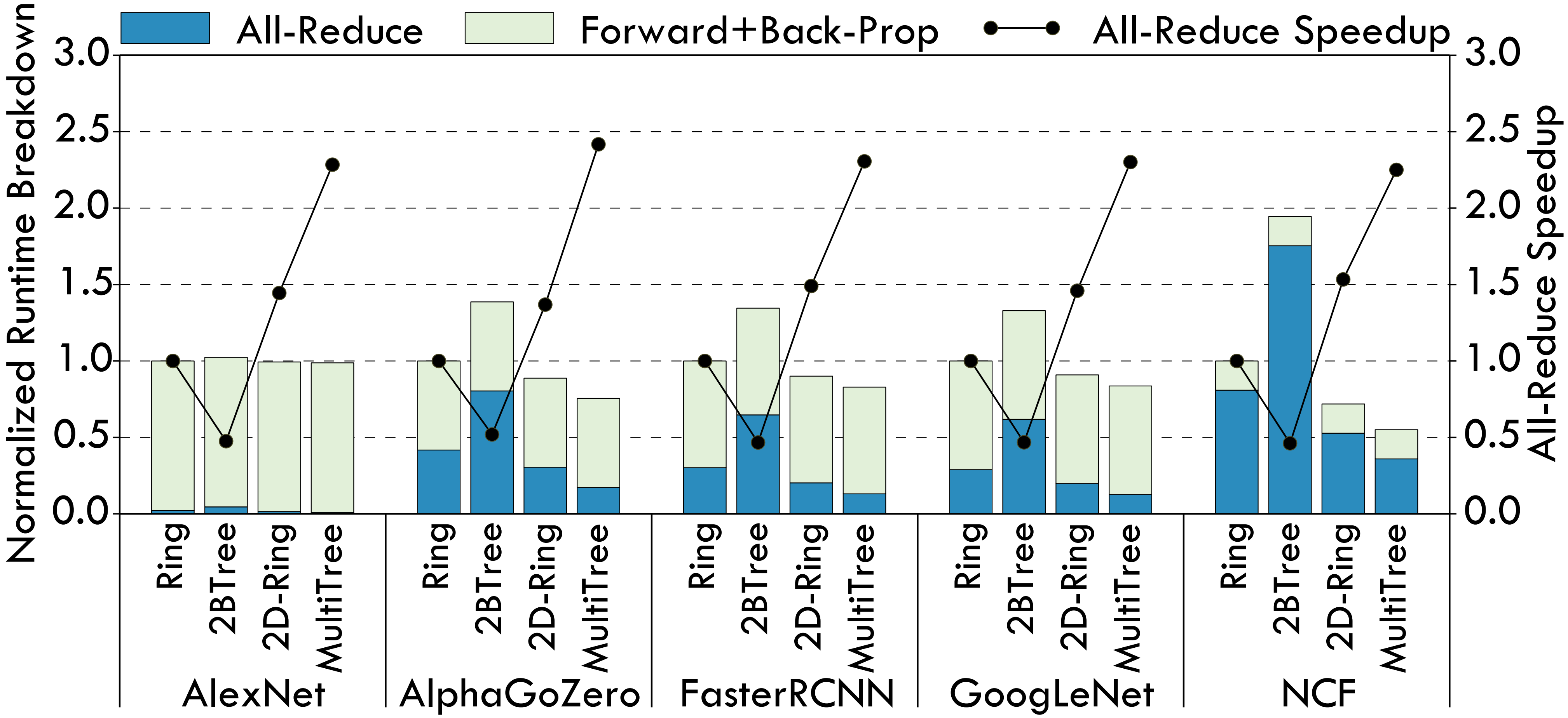
- Ring achieves good bandwidth for large data (long latency for small data)
- Double binary tree has good latency for small data, but bad at bandwidth

Results – Bandwidth on Switch-based Networks

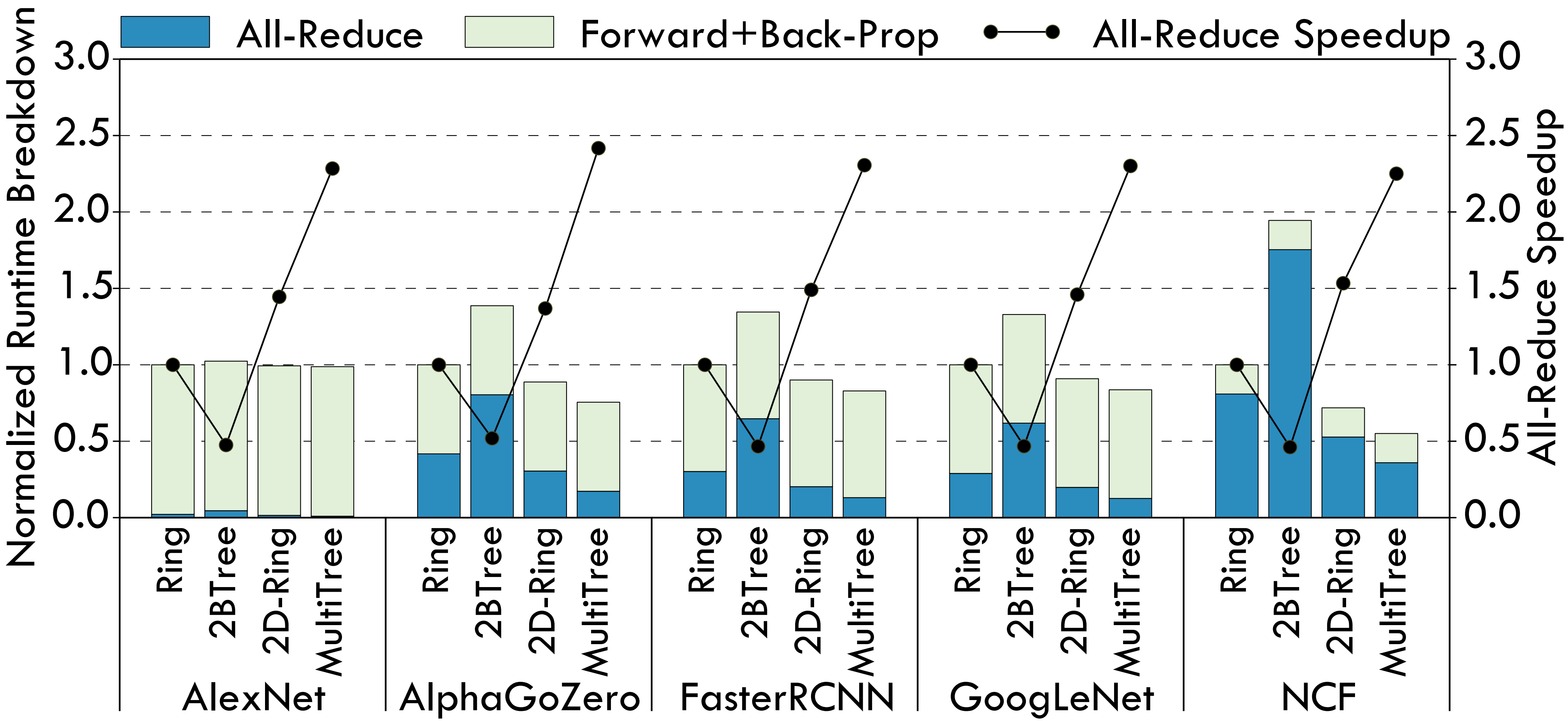


- Ring achieves good bandwidth for large data (long latency for small data)
- Double binary tree has good latency for small data, but bad at bandwidth
- MultiTree works well for both small and large data

Results – DNN Benchmarks on 8x8 Torus



Results – DNN Benchmarks on 8x8 Torus



- MultiTree > 2D-Ring > Ring > 2BTree (Double binary tree)

Summary

- Identifying inefficiencies in existing all-reduce algorithms

- MultiTree All-Reduce: algorithm-architecture co-design
  - Topology-aware and link usage coordination
  - Hardware-based all-reduce scheduling
  - Big message flow control for big gradients

- Achieves low latency as well as high throughput
  - Beats prior work with 2.5x improvement compared to Ring all-reduce