Basic Communication Operations (cont.)

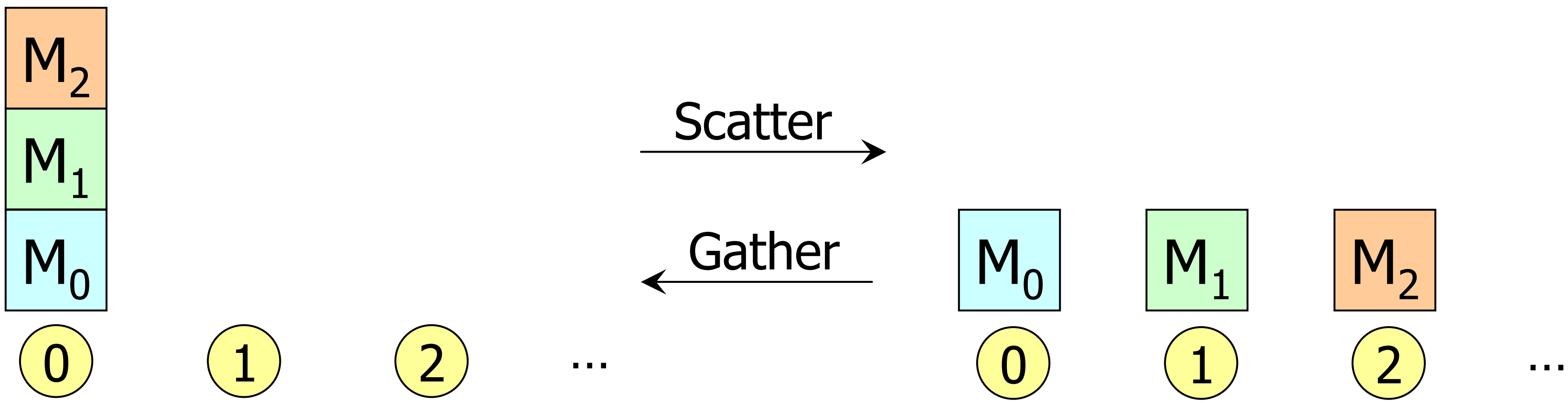
Alexandre David
B2-206

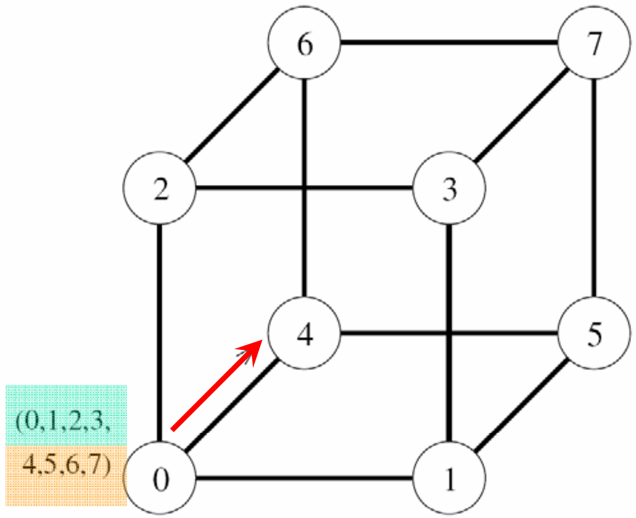
Today

- Scatter and Gather (4.4).
- All-to-All Personalized Communication (4.5).
- Circular Shift (4.6).
- Improving the Speed of Some Communication Operations (4.7).

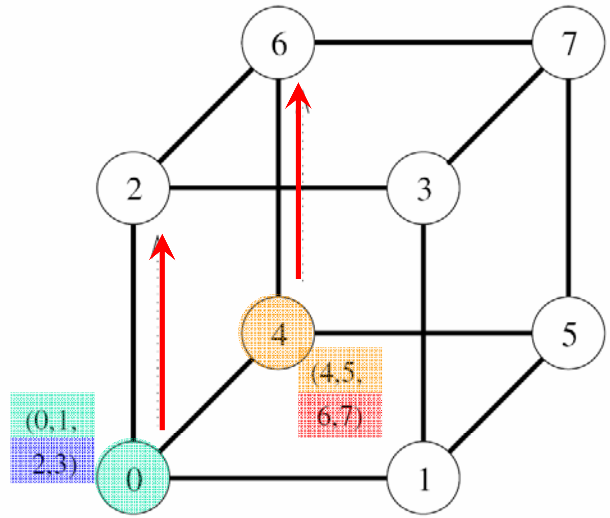
Scatter and Gather

- Scatter: A node sends a unique message to every other node – *unique per node*.
- Gather: Dual operation but the target node does not combine the messages into one.

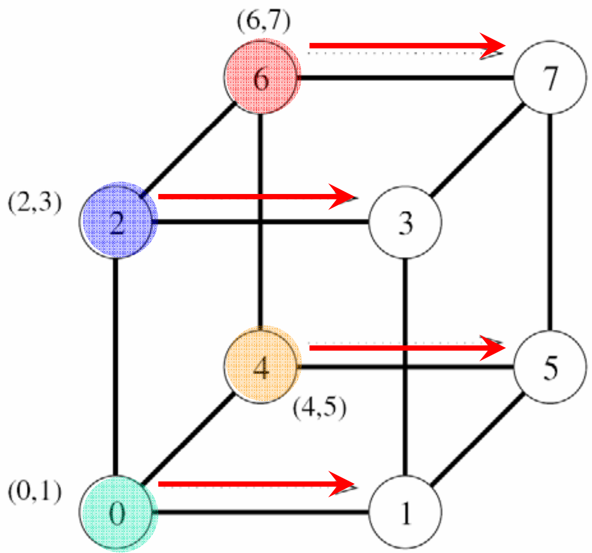




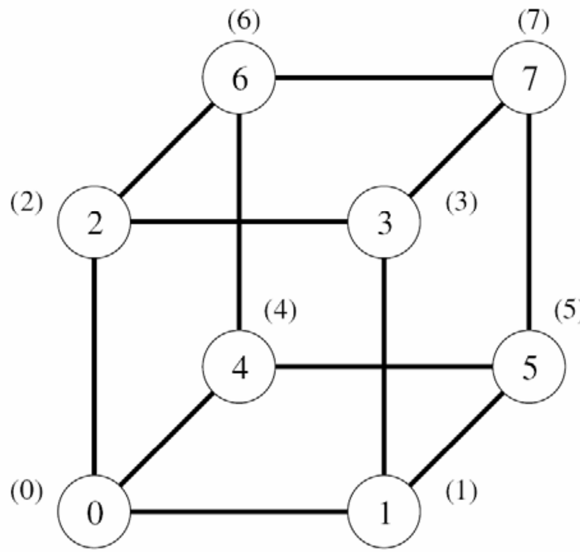
(a) Initial distribution of messages



(b) Distribution before the second step



(c) Distribution before the third step



(d) Final distribution of messages

Cost Analysis

- Number of steps: log*p*.
- Size transferred: *pm/2, pm/4,…,m*.
  - Geometric sum

$$p+\frac{p}{2}+\frac{p}{4}+\ldots+\frac{p}{2^n}=p\frac{1-\frac{1}{2^{n+1}}}{1-\frac{1}{2}}$$

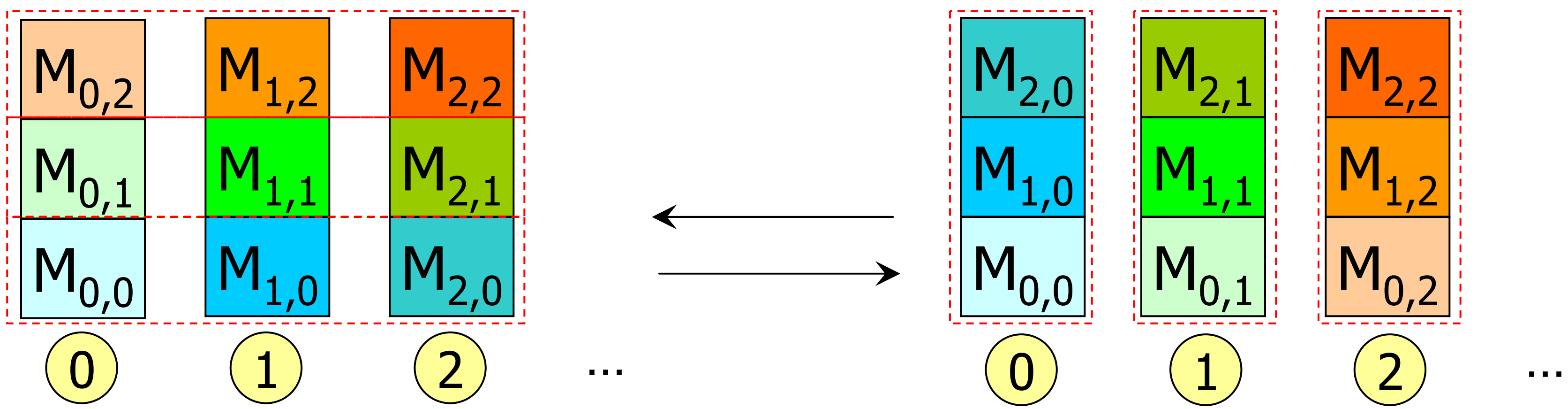
$$\frac{p}{2}+\frac{p}{4}+\ldots+\frac{p}{2^n}=2p(1-\frac{1}{2^{n+1}})-p=2p(1-\frac{1}{2p})-p=p-1$$

$$(2^{n+1}=2^{1+\log p}=2p)$$

- Cost $T=t_s \log p+t_w m(p-1)$.

All-to-All Personalized Communication

- Each node sends a *distinct* message to every other node.



Example: Transpose

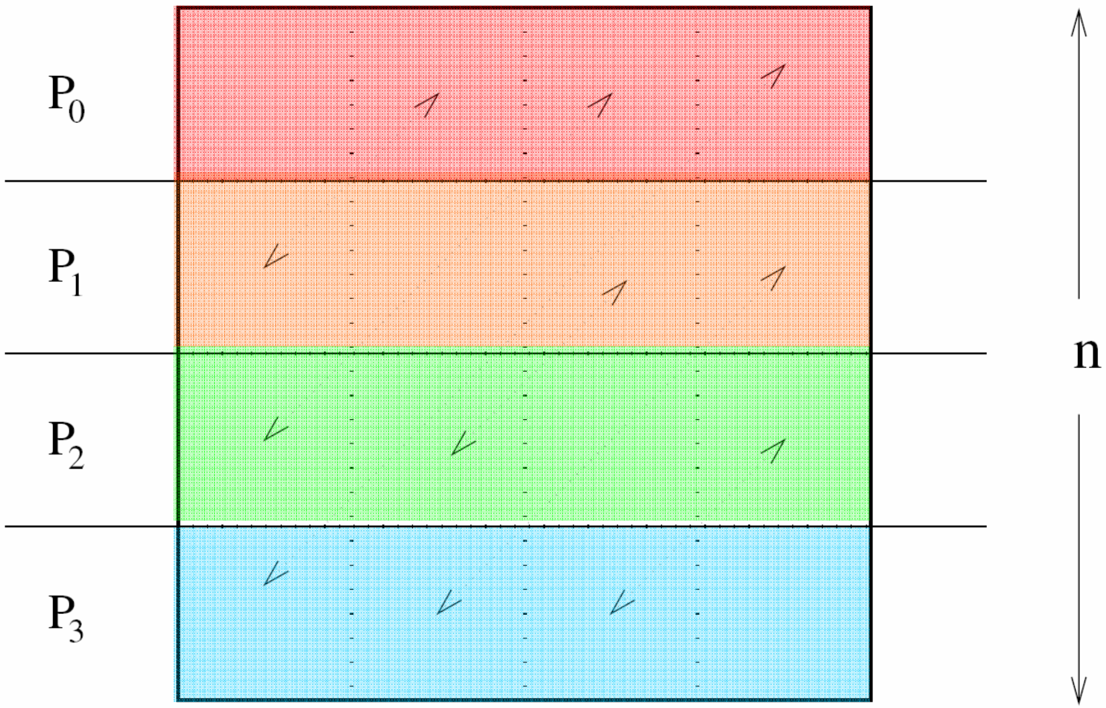
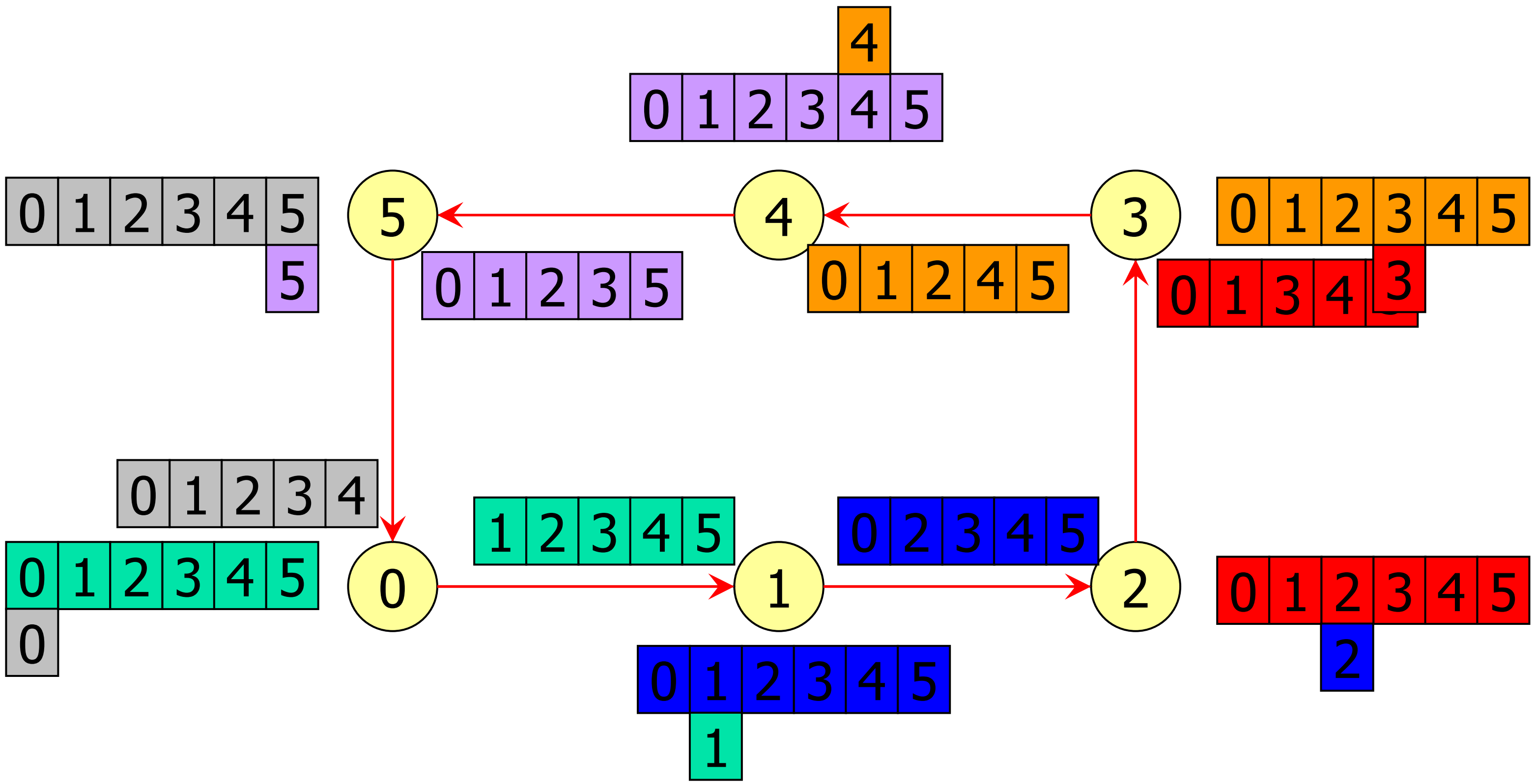
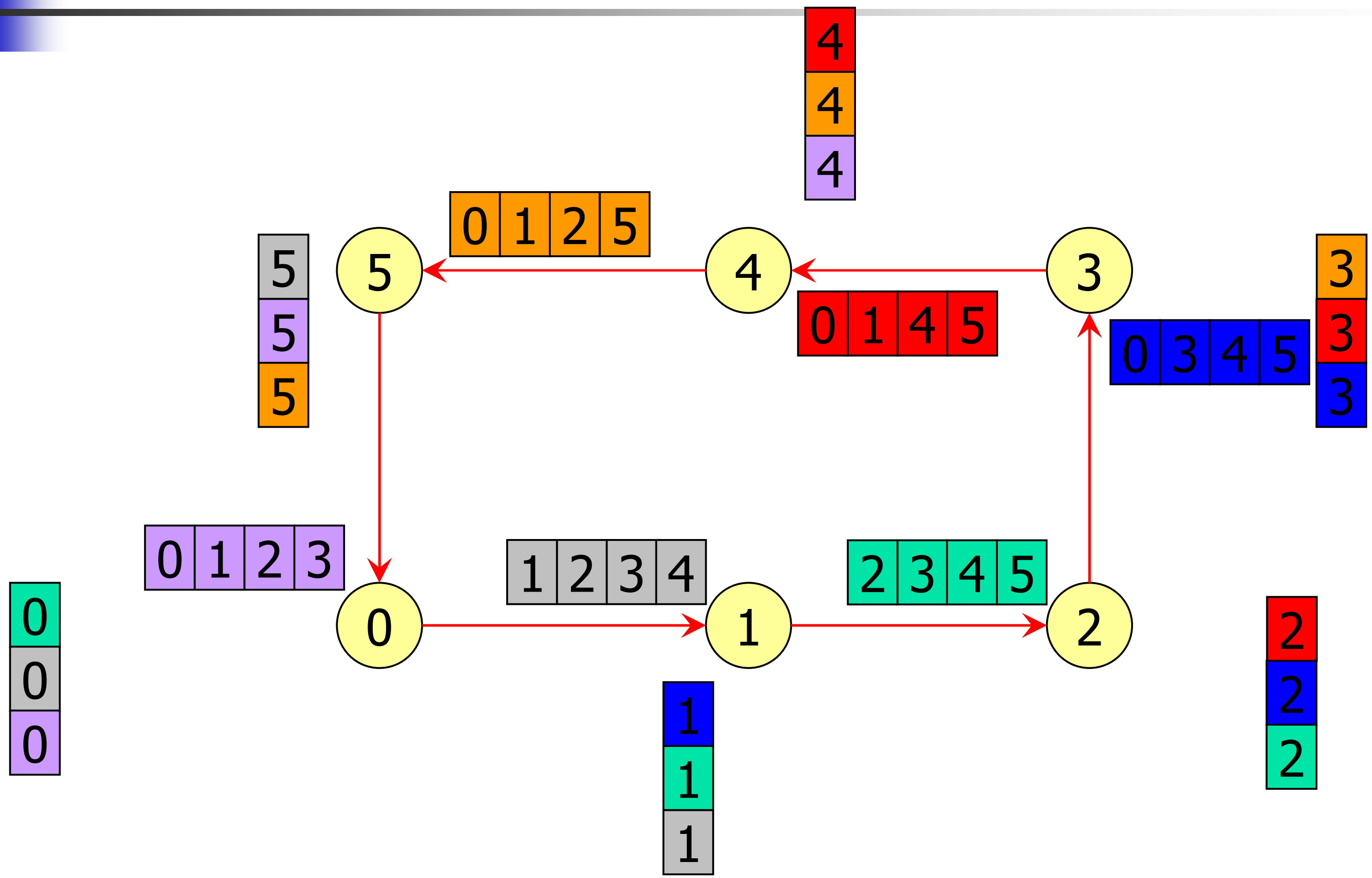


Figure 4.17 All-to-all personalized communication in transposing a 4×4 matrix using four processes.

Total Exchange on a Ring



Total Exchange on a Ring



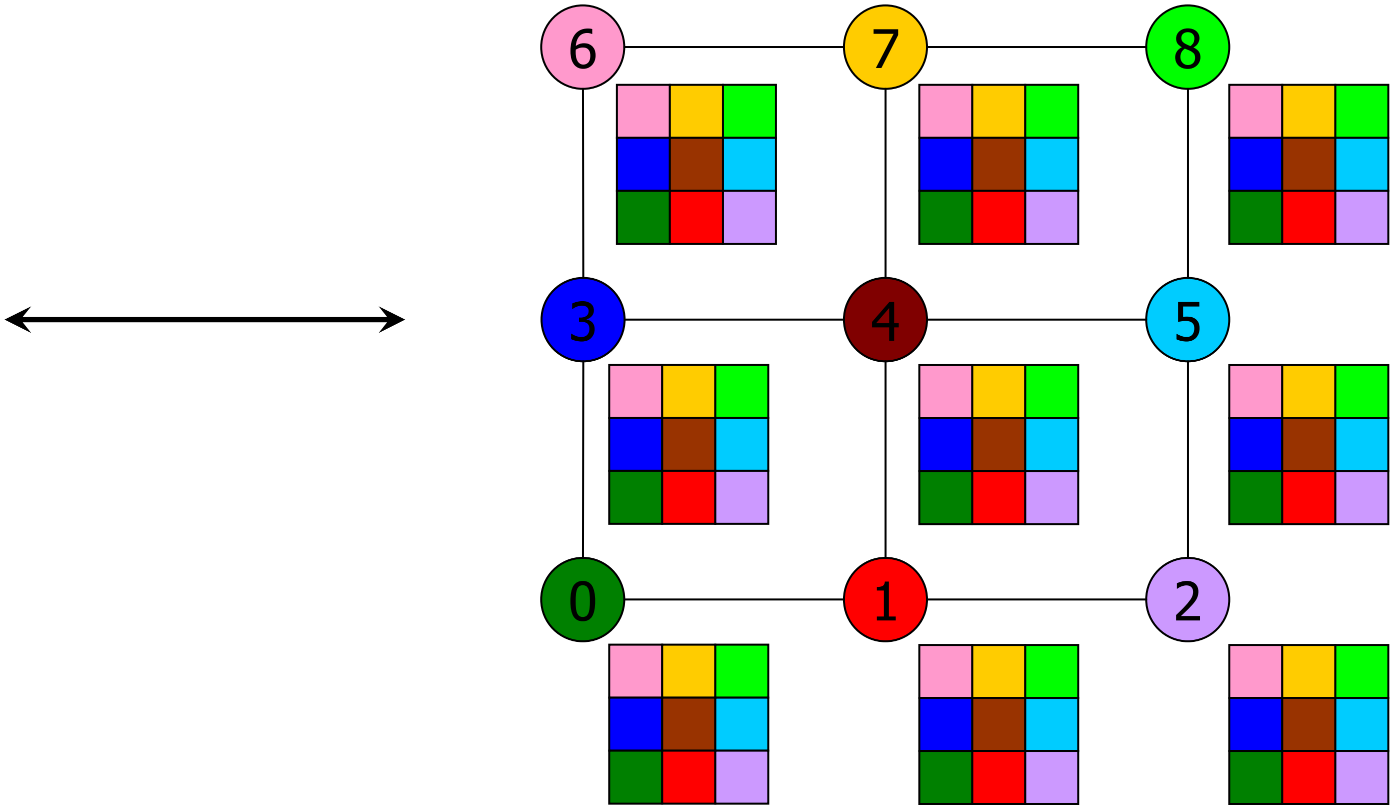
Cost Analysis

- Number of steps: *p-1*.
- Size transmitted: *m(p-1),m(p-2)…,m*.

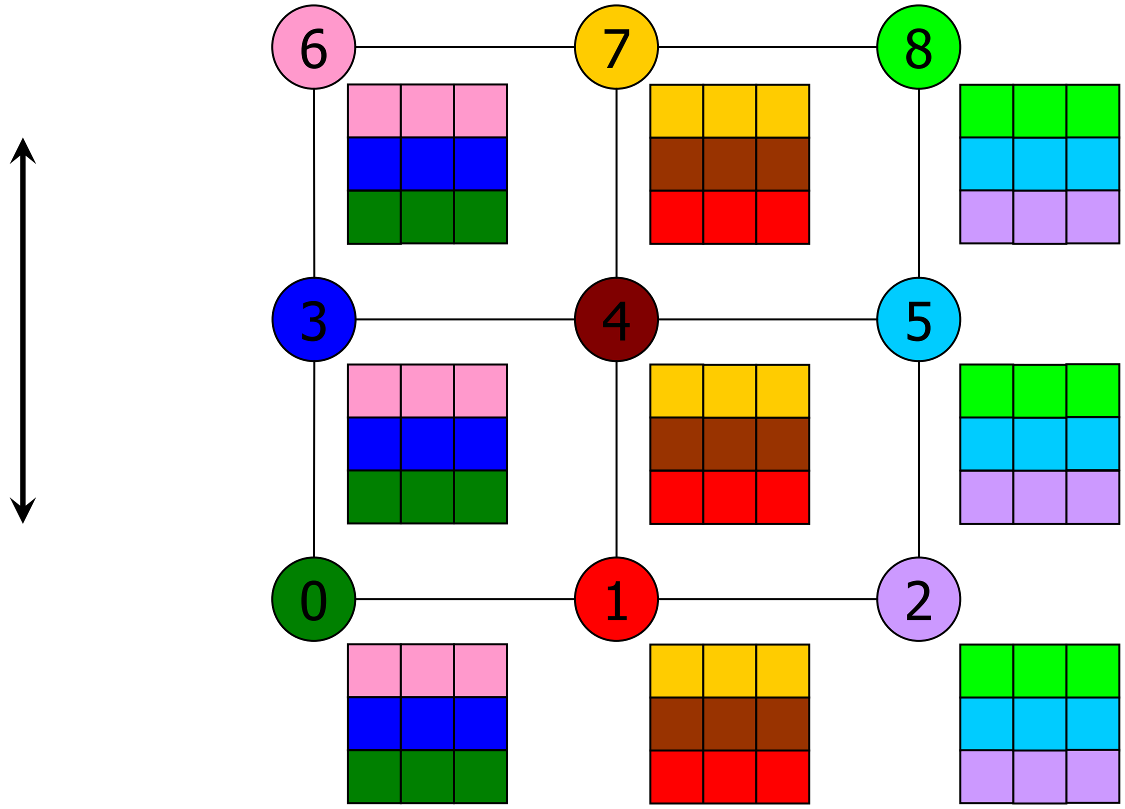
$$T = t_s(p-1) + \sum_{i=1}^{p-1} i t_w m = (t_s + t_w mp/2)(p-1)$$

Optimal

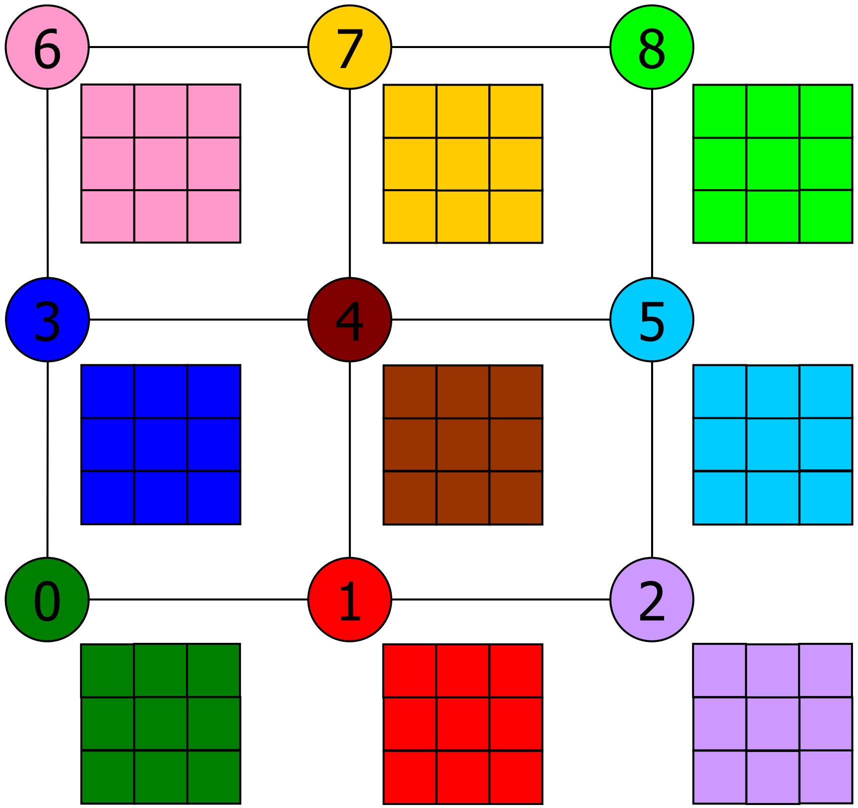
Total Exchange on a Mesh



Total Exchange on a Mesh



Total Exchange on a Mesh



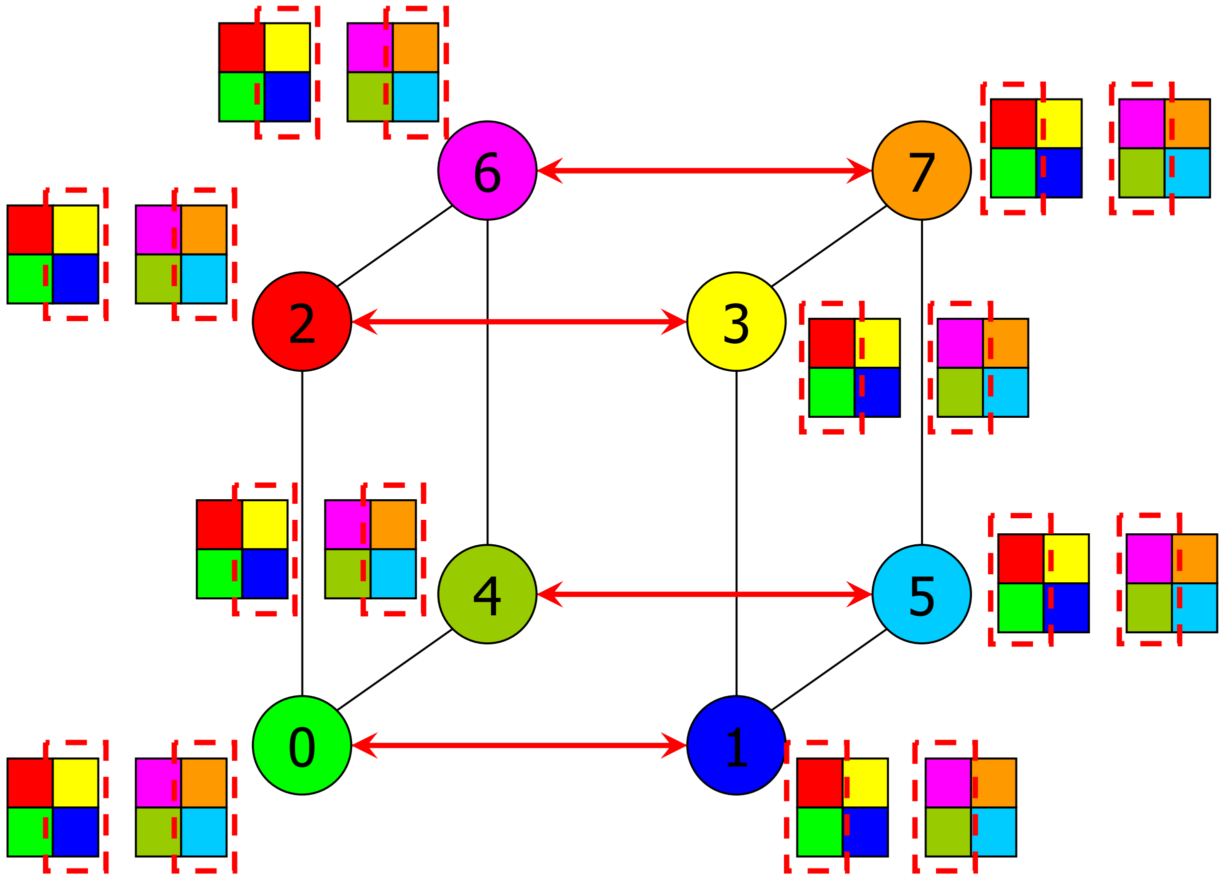
Cost Analysis

- Substitute p by $\sqrt{p}$ (number of nodes per dimension).
- Substitute message size m by $m\sqrt{p}$.
- Cost is the same for each dimension.
- $T=(2t_s+t_w mp)(\sqrt{p}-1)$

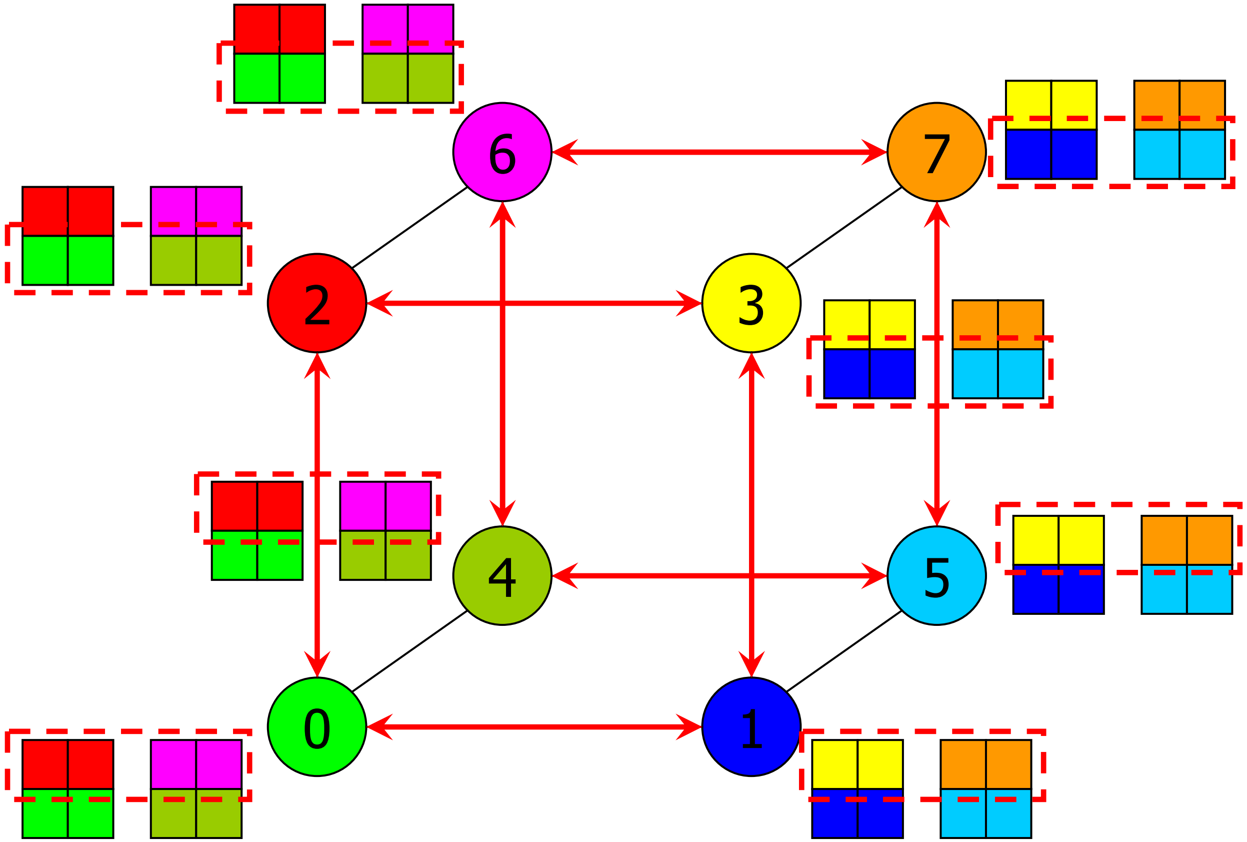
Total Exchange on a Hypercube

- Generalize the mesh algorithm to log*p* steps = number of dimensions, with 2 nodes per dimension.
- Same procedure as all-to-all broadcast.

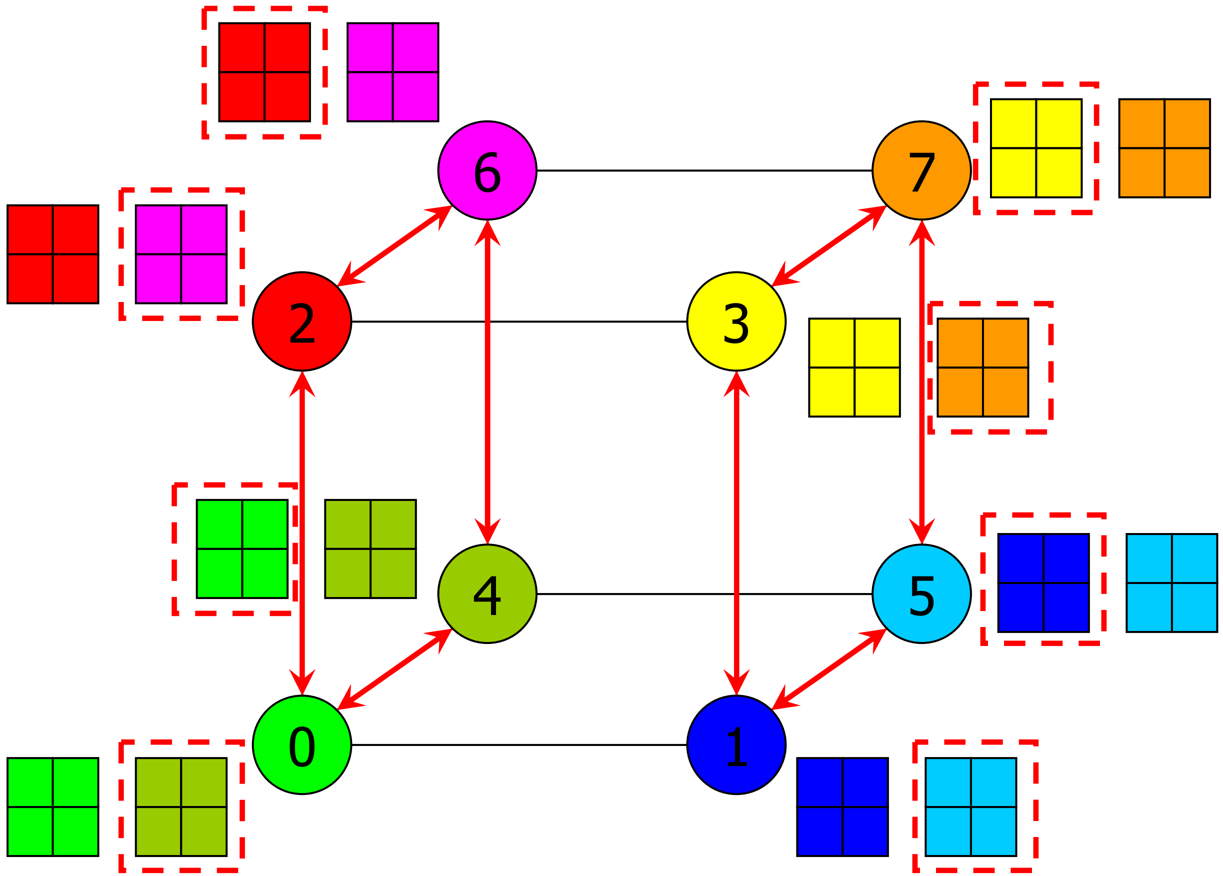
Total Exchange on a Hypercube



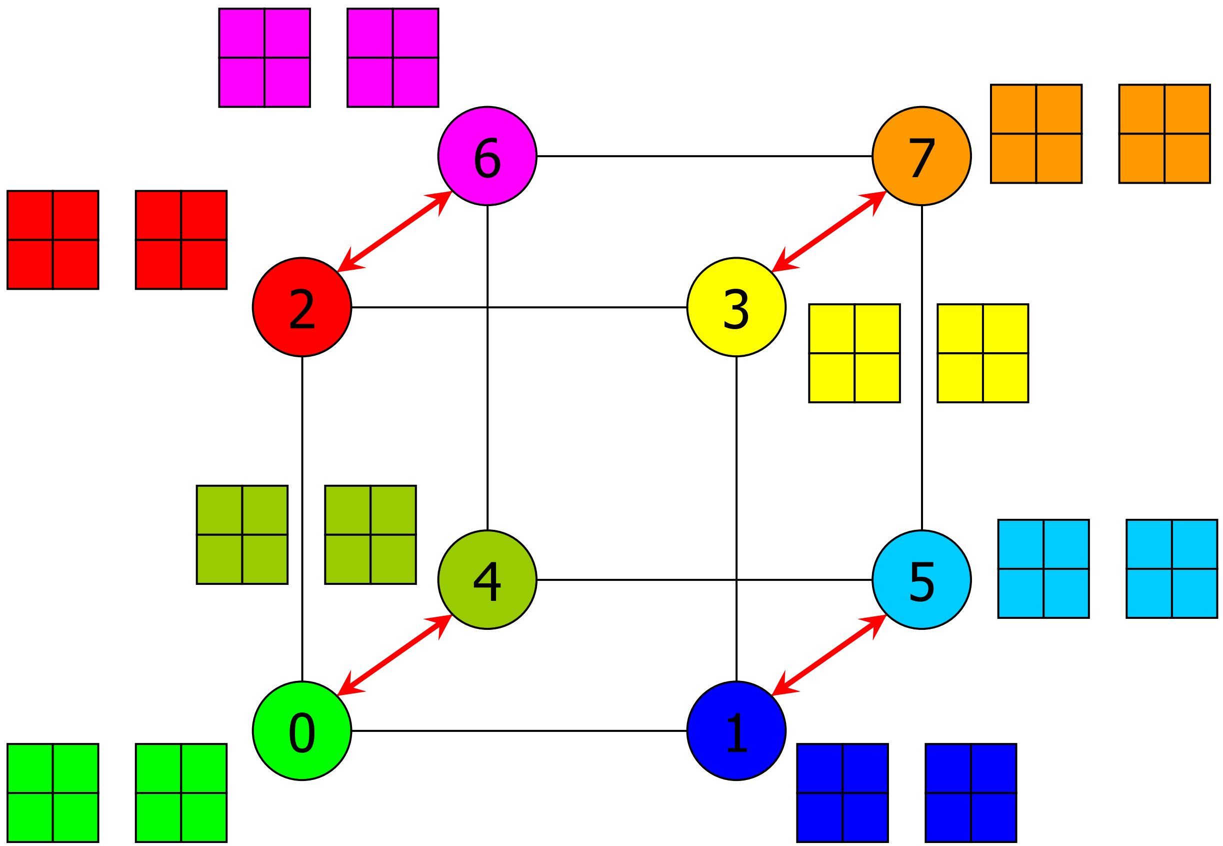
Total Exchange on a Hypercube



Total Exchange on a Hypercube



Total Exchange on a Hypercube



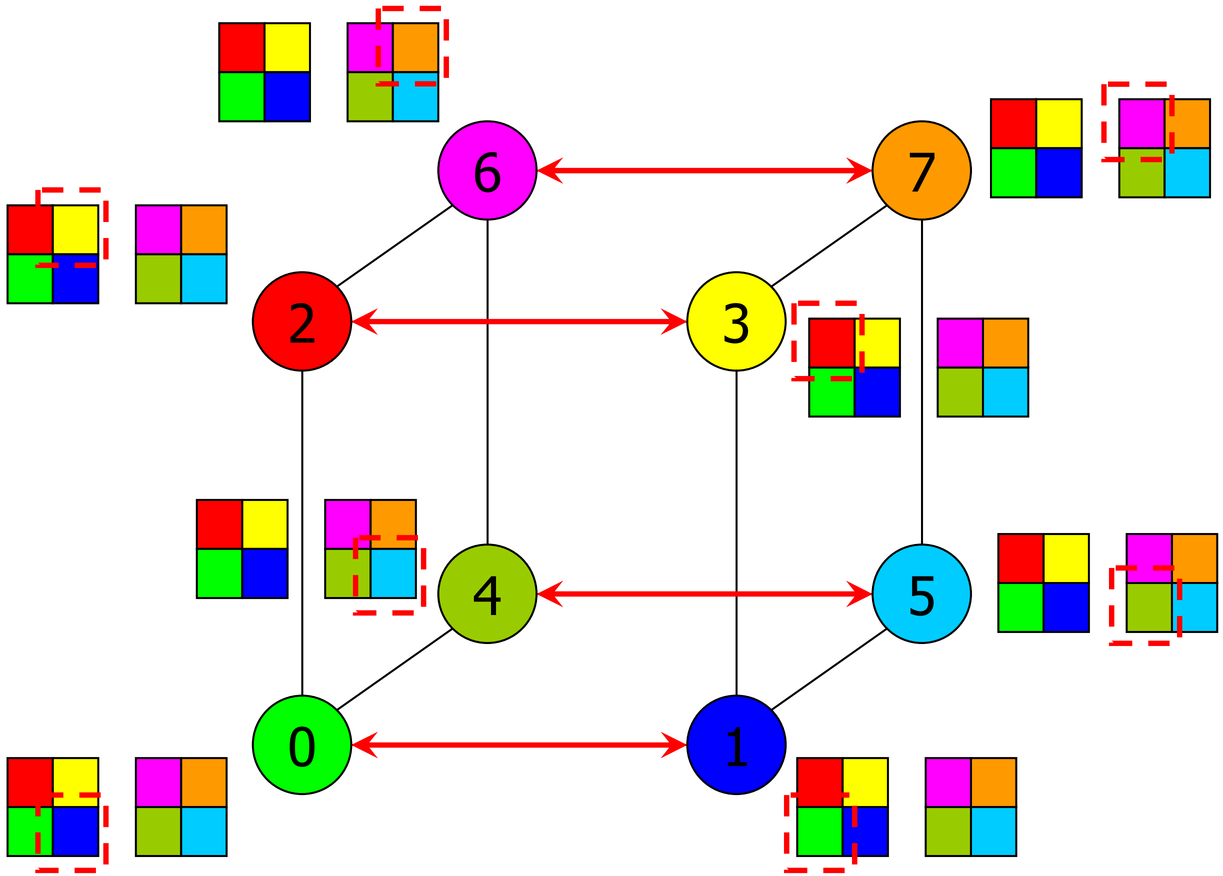
Cost Analysis

- Number of steps: log*p*.
- Size transmitted per step: *pm/2*.
- Cost: $T=(t_s+t_w mp/2)\log p$.
- Optimal? NO
- Each node sends and receives m(p-1) words. Average distance = (log*p*)/2. Total traffic = $p*m(p-1)*\log p/2$.
- Number of links = $p \log p/2$.
- Time lower bound = $t_w m(p-1)$.

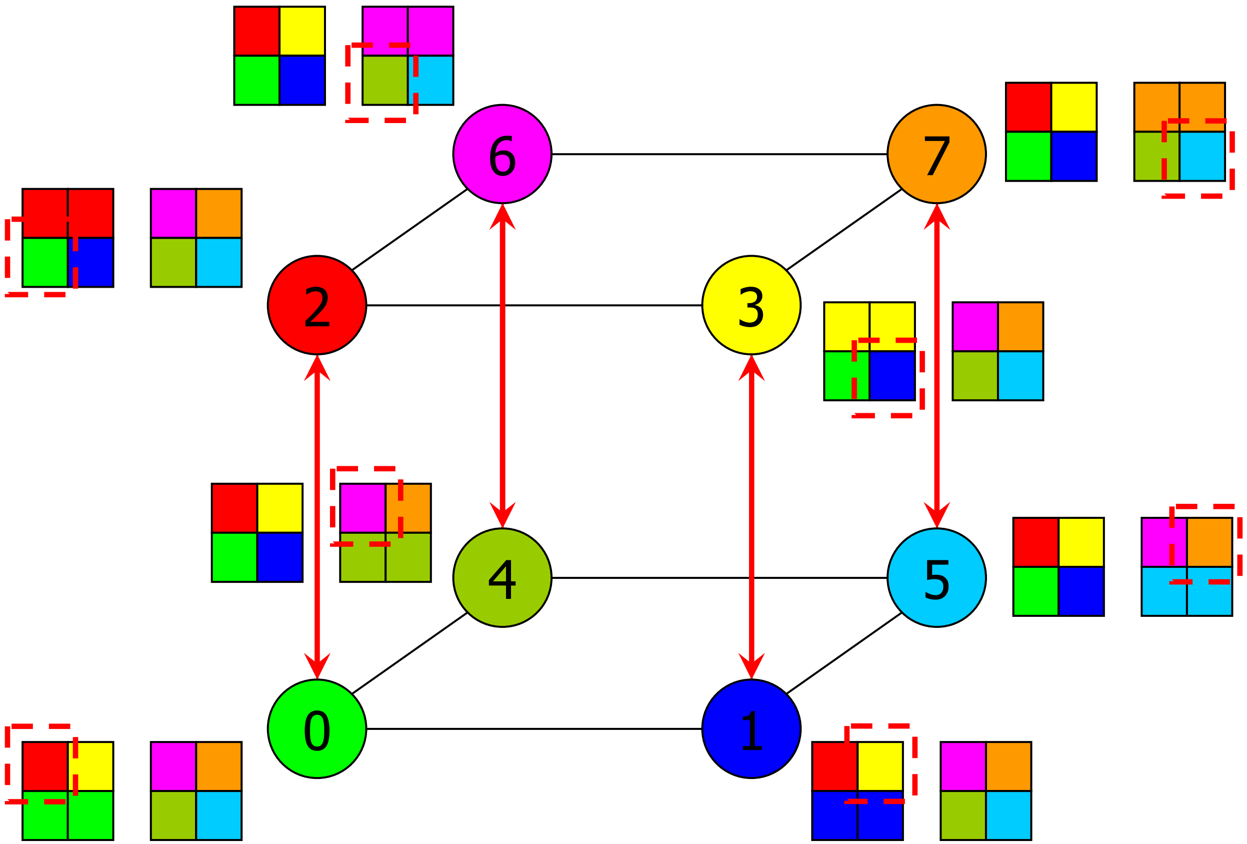
An Optimal Algorithm

- Have every pair of nodes communicate directly with each other – p-1 communication steps – but without congestion.
- At j^{th} step node i communicates with node $(i \text{ xor } j)$ with E-cube routing.

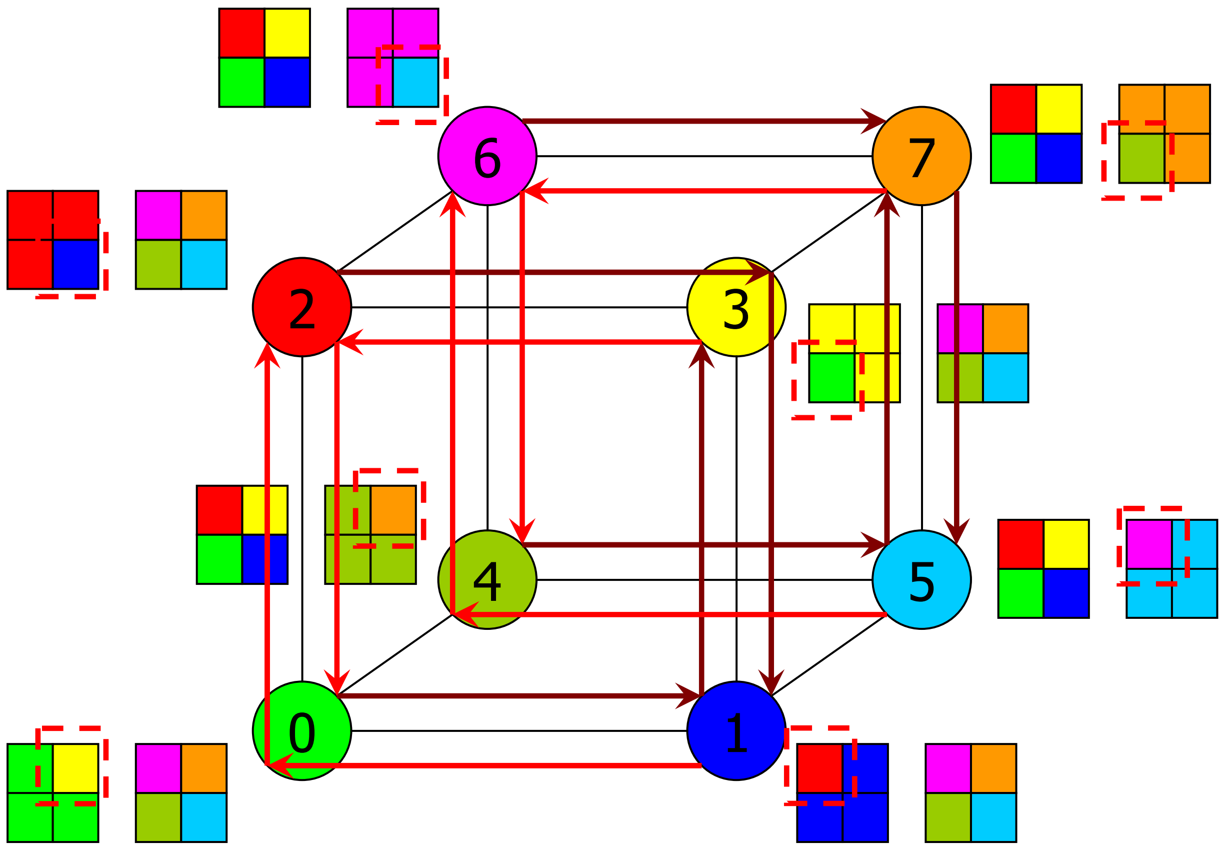
Total Exchange on a Hypercube



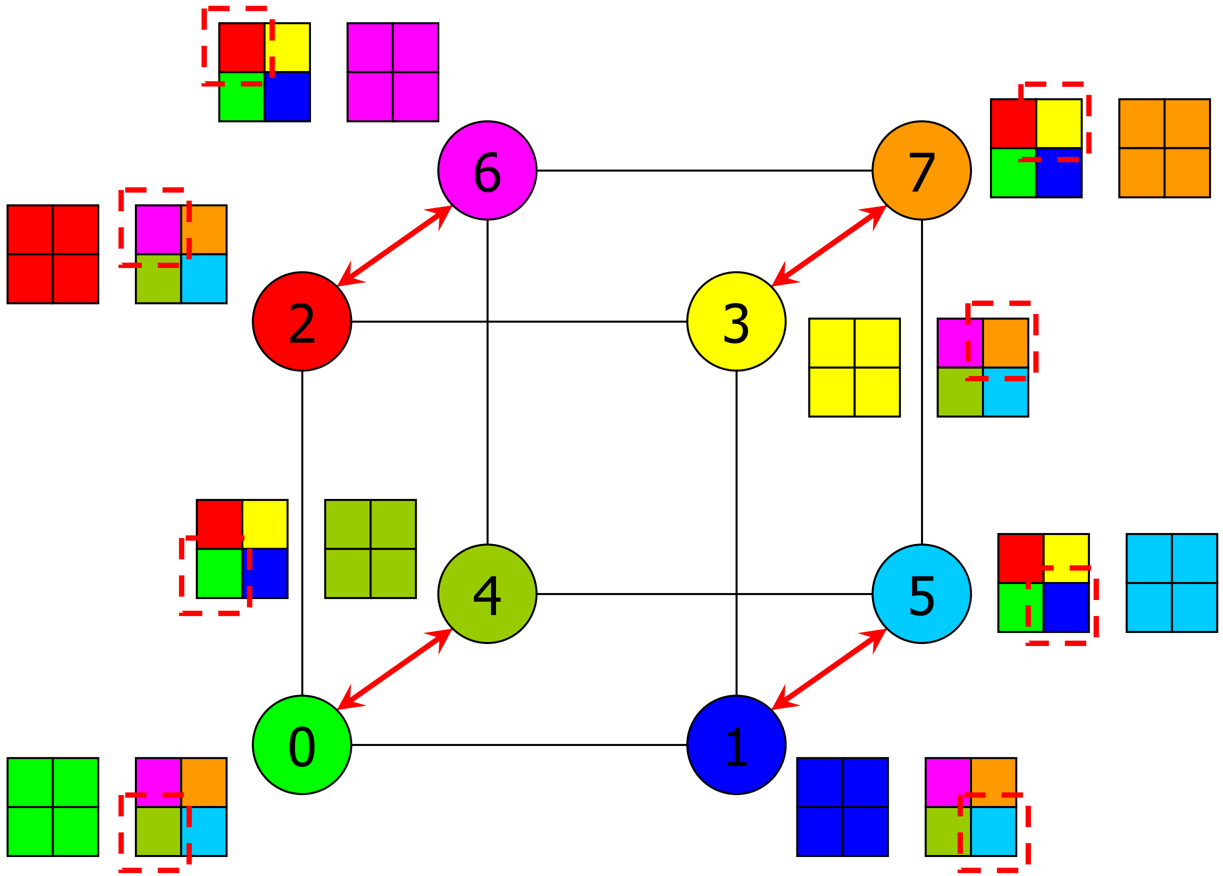
Total Exchange on a Hypercube



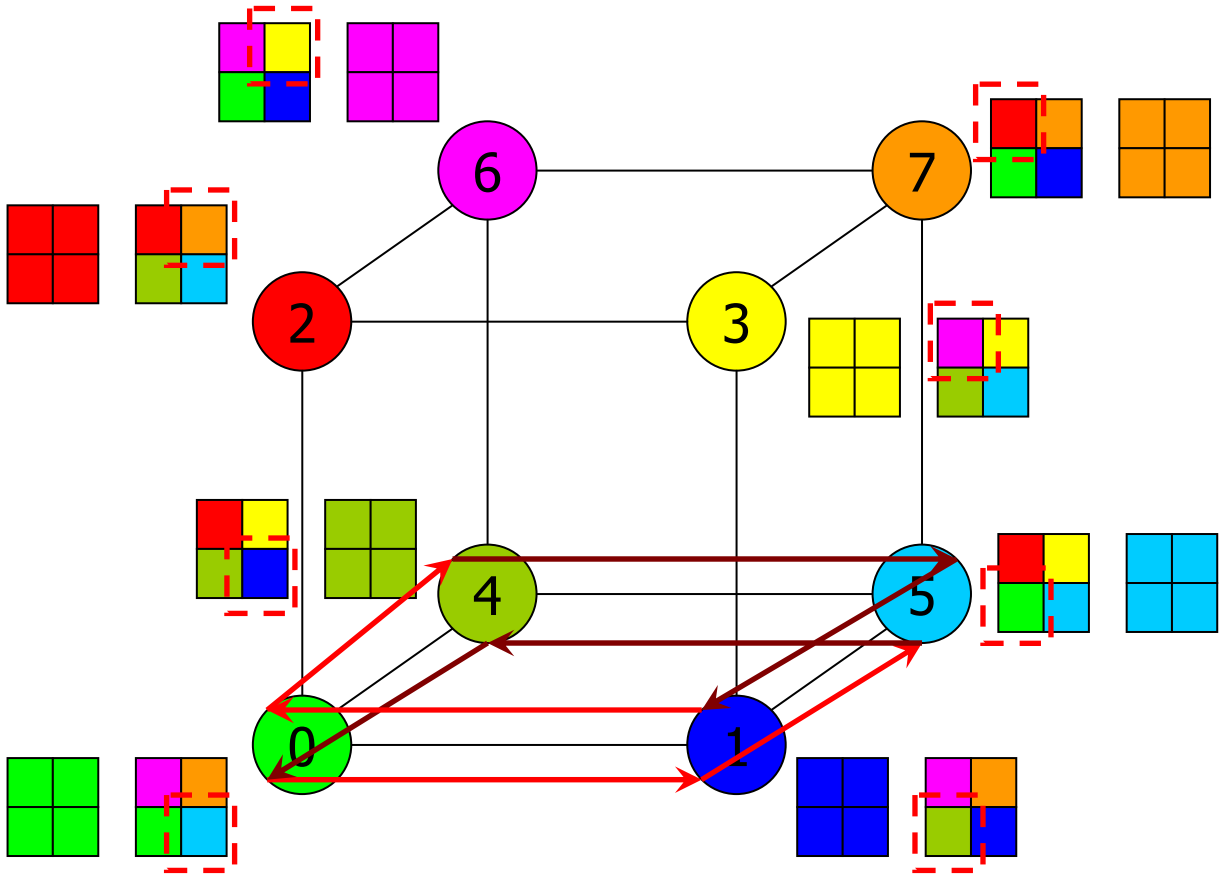
Total Exchange on a Hypercube



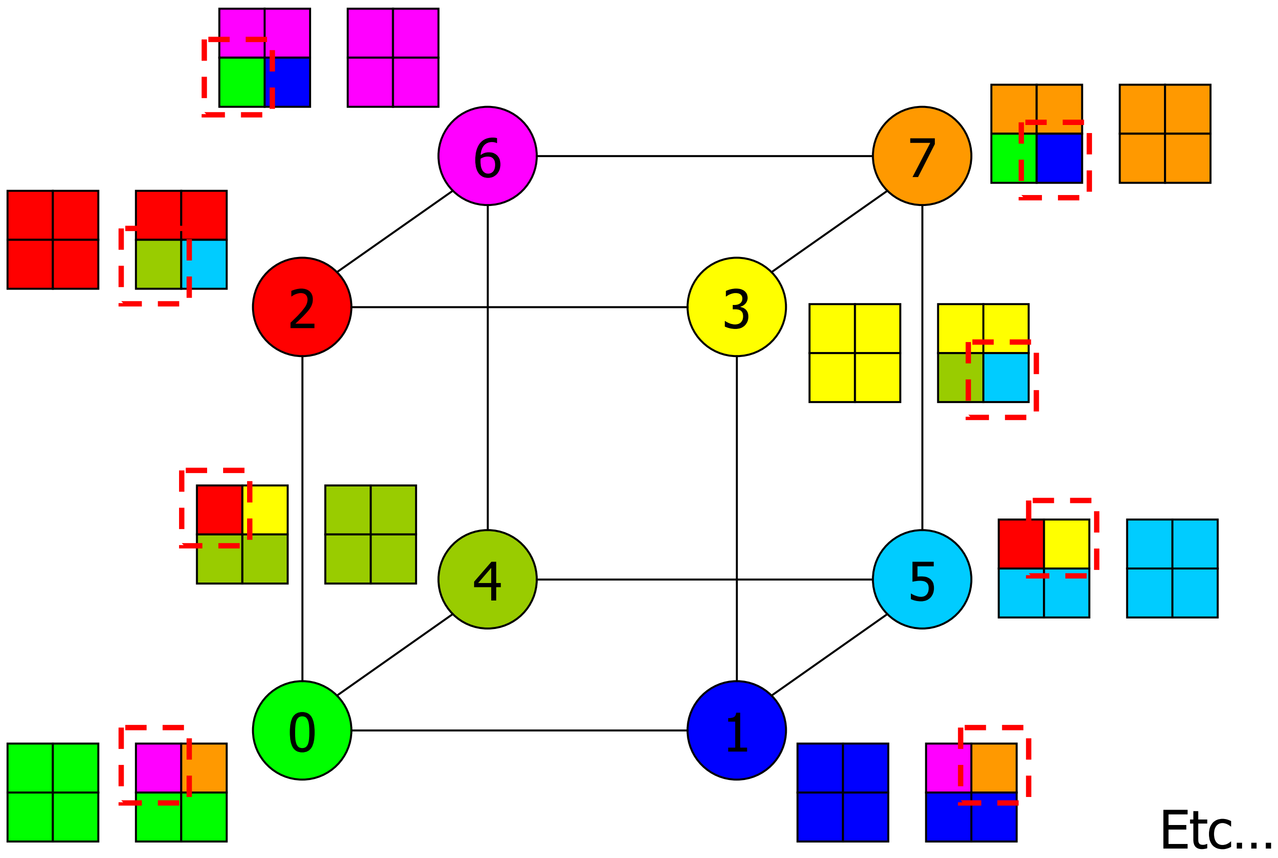
Total Exchange on a Hypercube



Total Exchange on a Hypercube



Total Exchange on a Hypercube



Cost Analysis

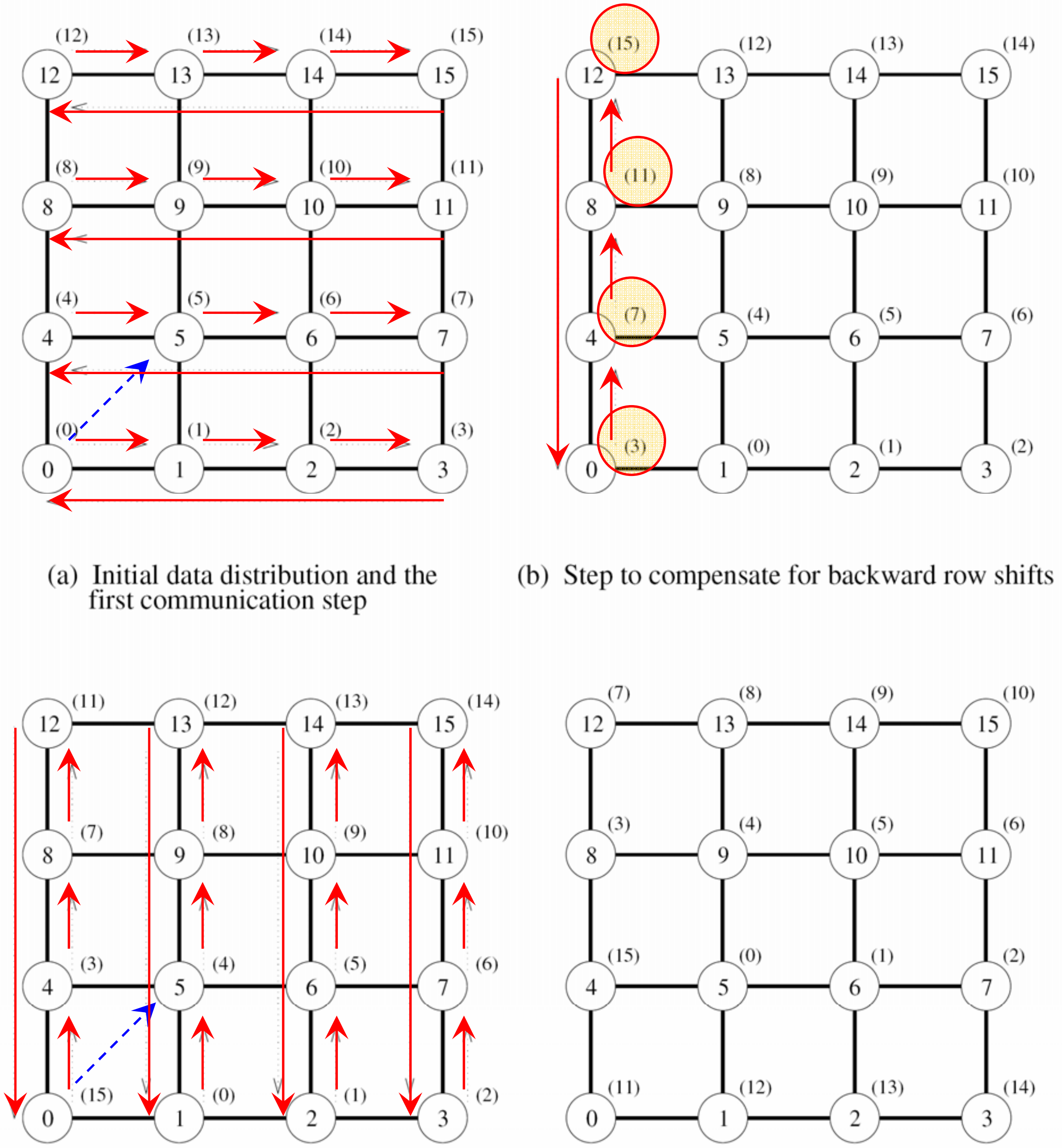
- Remark: Transmit less, only what is needed, but more steps.
- Number of steps: $p-1$.
- Transmission: size m per step.
- Cost: $T=(t_s+t_w m)(p-1)$.
- Compared with $T=(t_s+t_w mp/2)\log p$.
- Previous algorithm better for small messages.

Circular Shift

- It's a particular permutation.
- Circular q-shift: Node *i* sends data to node *(i+q) mod p* (in a set of p nodes).
- Useful in some matrix operations and pattern matching.
- Ring: intuitive algorithm in min{q,p-q} neighbor to neighbor communication steps. Why?

Circular 5-shift on a mesh.

q mod √p on rows compensate
⌊q / √p⌋ on colums



(a) Initial data distribution and the first communication step

(b) Step to compensate for backward row shifts

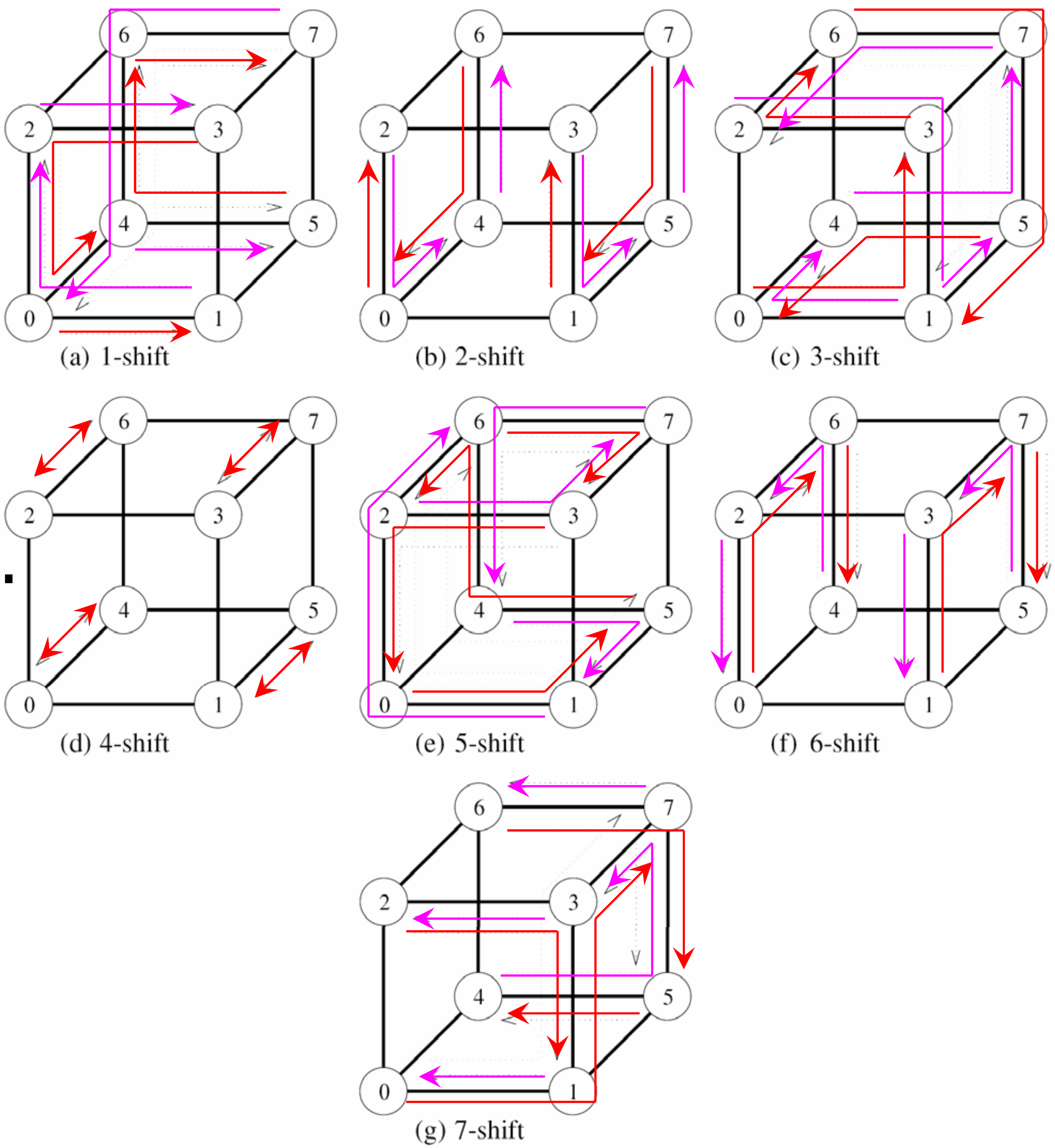
(c) Column shifts in the third communication step

(d) Final distribution of the data

Circular Shift on a Hypercube

- Map a linear array with 2^d nodes onto a hypercube of dimension d.
- Expand q shift as a sum of powers of 2 (e.g. 5-shift = 2^0+2^2).
- Perform the decomposed shifts.
- Use bi-directional links for "forward" (shift itself) and "backward" (rotation part)… $\log p$ steps.

Or better:
Direct
E-cube routing.
q-shifts on a
8-node
hypercube.



Improving Performance

- So far messages of size *m* were not split.
- If we split them into *p* parts:
  - One-to-all broadcast = scatter + all-to-all broadcast of messages of size *m/p*.
  - All-to-one reduction = all-to-all reduce + scatter of messages of size *m/p*.
  - All-reduce = all-to-all reduction + all-to-all broadcast of messages of size *m/p*.