Parallel and Distributed Computing

CS3006

Lecture 12

Basic Communication Operations-II

20th April 2022

Dr. Rana Asif Rehman

All-to-All Broadcast and All-to-All Reduction

Basic Communication Operations

(All-to-All Broadcast and All-to-All Reduction)

All-to-All Broadcast

- A generalization to of one-to-all broadcast.
- Every process broadcasts m-word message.
  - The broadcast-message for each of the processes can be different than others

All-to-All Reduction

- Dual of all-to-all broadcast
- Each node is the destination of an all-to-one reduction out of total P reductions.

Basic Communication Operations

(All-to-All Broadcast and All-to-All Reduction)

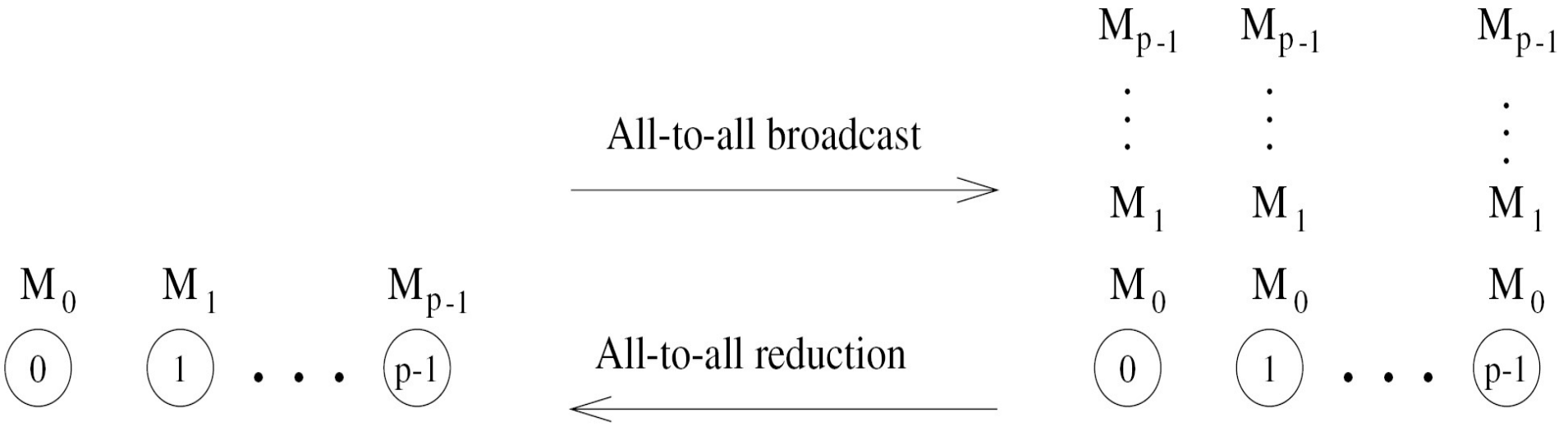


Figure 4.8 All-to-all broadcast and all-to-all reduction.

A naïve Broadcast method may be performing **P** one-to-all broadcasts. This will result **P(log(p)(t(s) + mt(w)))** communication time.

Solution?

Linear Ring Broadcast

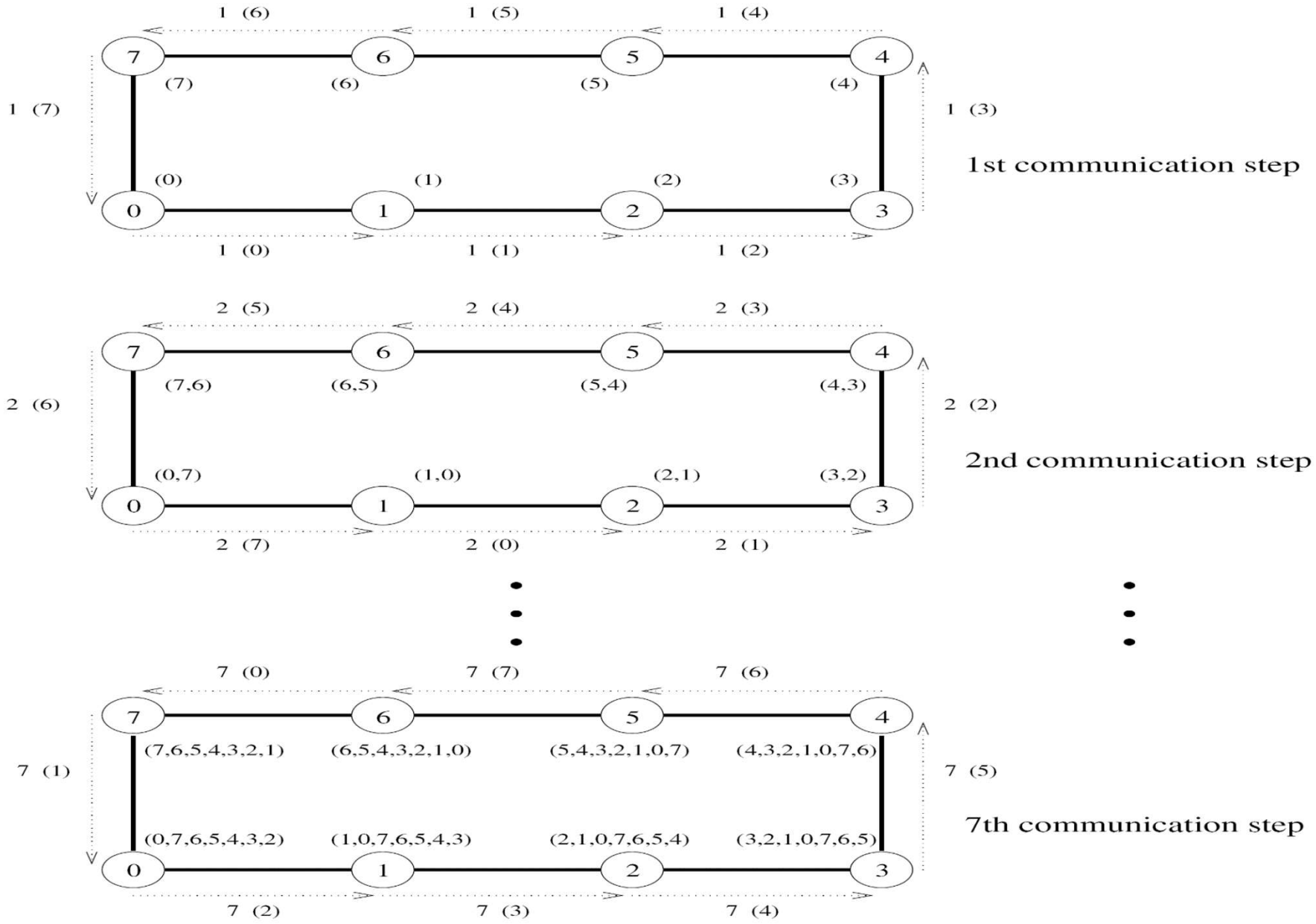


Figure 4.9 All-to-all broadcast on an eight-node ring. The label of each arrow shows the time step and, within parentheses, the label of the node that owned the current message being transferred before the beginning of the broadcast. The number(s) in parentheses next to each node are the labels of nodes from which data has been received prior to the current communication step. Only the first, second, and last communication steps are shown.

Linear Ring Broadcast


```
1.   procedure ALL_TO_ALL_BC_RING(my_id, my_msg, p, result)
2.   begin
3.       left := (my_id − 1) mod p;
4.       right := (my_id + 1) mod p;
5.       result := my_msg;
6.       msg := result;
7.       for i := 1 to p − 1 do
8.           send msg to right;
9.           receive msg from left;
10.          result := result ∪ msg;
11.      endfor;
12.  end ALL_TO_ALL_BC_RING
```

Algorithm 4.4 All-to-all broadcast on a p-node ring.

Linear Ring Broadcast

Home Task (graded as extra bonus)

```
Enter number of processes in the Ring:4

PHASE: 1

0 is sending  0 to 1    0 is receiving 3 from 3
1 is sending  1 to 2    1 is receiving 0 from 0
2 is sending  2 to 3    2 is receiving 1 from 1
3 is sending  3 to 0    3 is receiving 2 from 2

PHASE: 2

0 is sending  3 to 1    0 is receiving 2 from 3
1 is sending  0 to 2    1 is receiving 3 from 0
2 is sending  1 to 3    2 is receiving 0 from 1
3 is sending  2 to 0    3 is receiving 1 from 2

PHASE: 3

0 is sending  2 to 1    0 is receiving 1 from 3
1 is sending  3 to 2    1 is receiving 2 from 0
2 is sending  0 to 3    2 is receiving 3 from 1
3 is sending  1 to 0    3 is receiving 0 from 2

Data Accumulated in P0  is:
3       2       1        0
Data Accumulated in P1  is:
0       3       2        1
Data Accumulated in P2  is:
1       0       3        2
Data Accumulated in P3  is:
2       1       0        3

--------------------------------
Process exited after 57.51 seconds with return value 0
Press any key to continue . . .
```

Basic Communication Operations

(All-to-All Broadcast and All-to-All Reduction)

Linear Array or Ring

- **Reduction**
  - Draw an All-to-All Broadcast on a P-node linear ring
  - Reverse the directions in each foreach of the step without changing message
  - After each communication step, combine messages having same broadcast destination with associative operator.

- **Now, Its your turn to draw?**
  - **Draw an All-to-All Broadcast on a 4-node linear ring**
  - **Reverse the directions and combine the results using 'SUM'**

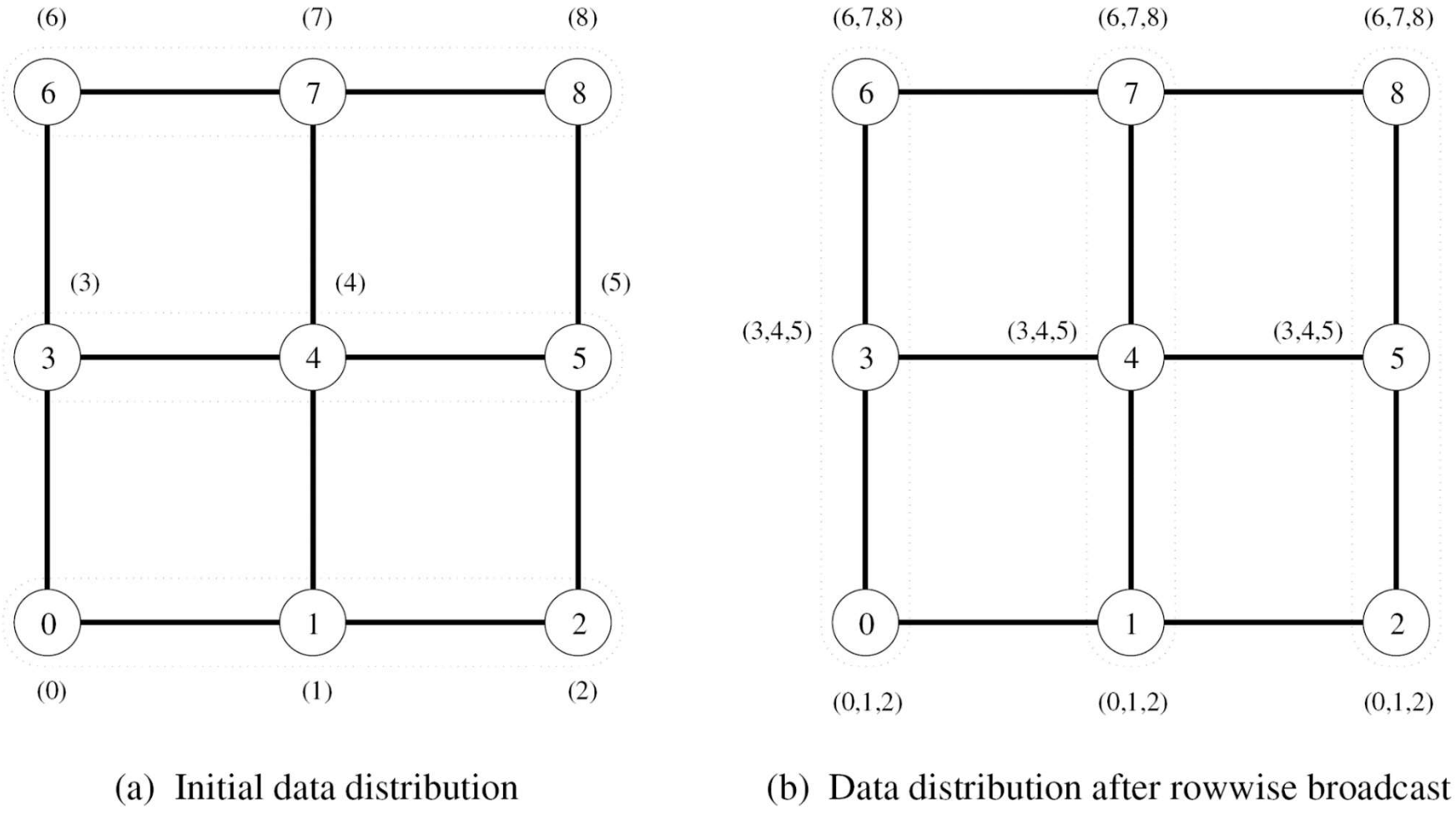
Linear Ring Reduction

```
1.   procedure ALL_TO_ALL_RED_RING(my_id, my_msg, p, result)
2.   begin
3.       left := (my_id − 1) mod p;
4.       right := (my_id + 1) mod p;
5.       recv := 0;
6.       for i := 1 to p − 1 do
7.           j := (my_id + i) mod p;
8.           temp := msg[j] + recv;
9.           send temp to left;
10.          receive recv from right;
11.      endfor;
12.      result := msg[my_id] + recv;
13.  end ALL_TO_ALL_RED_RING
```

Algorithm 4.5 All-to-all reduction on a p-node ring.

Basic Communication Operations

(All-to-All Broadcast on 2D Mesh..Algorithm 4.6)



(a) Initial data distribution

(b) Data distribution after rowwise broadcast

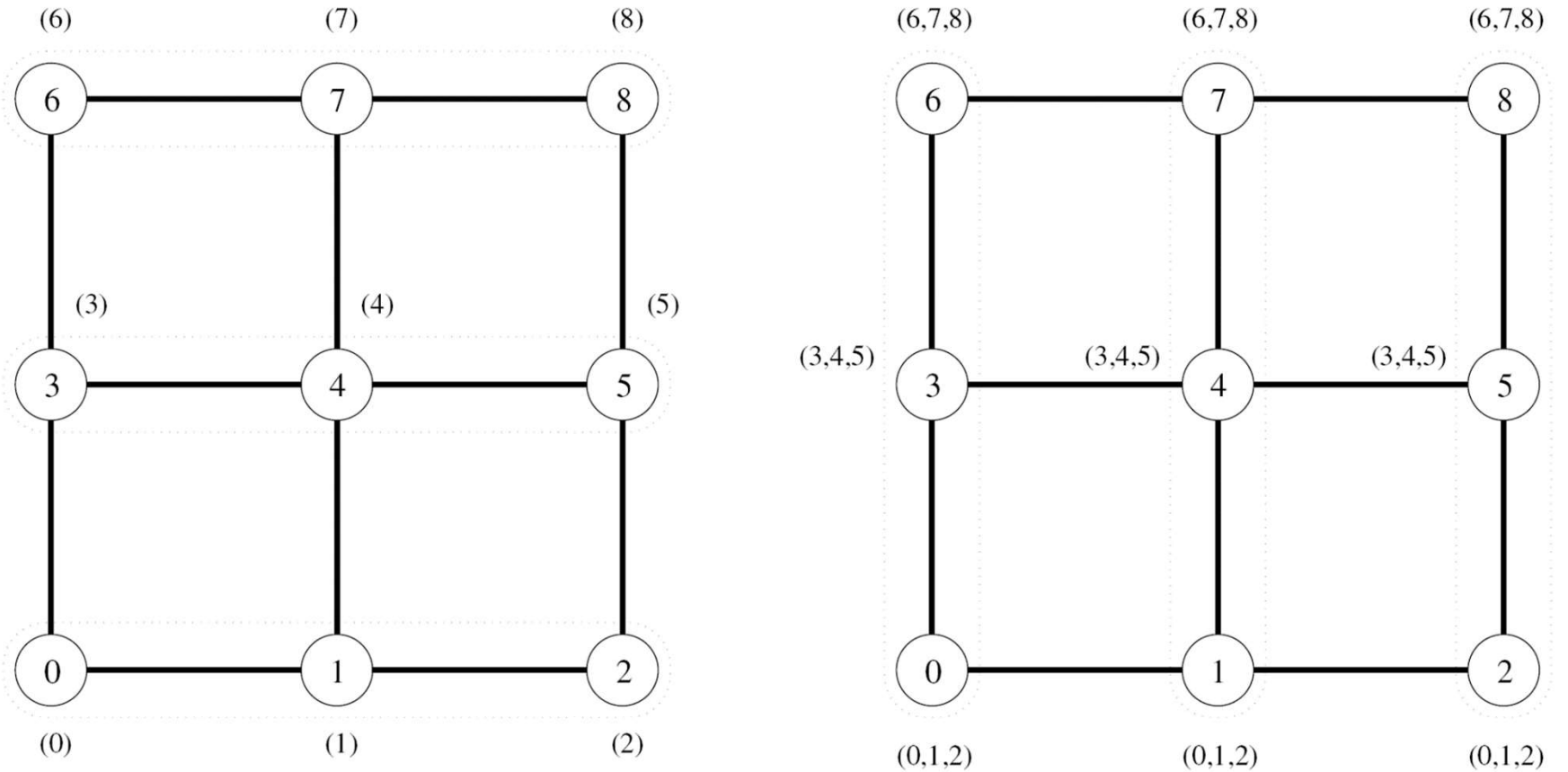
Figure 4.10 All-to-all broadcast on a 3×3 mesh. The groups of nodes communicating with each other in each phase are enclosed by dotted boundaries. By the end of the second phase, all nodes get (0,1,2,3,4,5,6,7) (that is, a message from each node).

```
1.    procedure ALL_TO_ALL_BC_MESH(my_id, my_msg, p, result)
2.    begin

/* Communication along rows */
3.        left := my_id − (my_id mod √p) + (my_id − 1) mod √p;
4.        right := my_id − (my_id mod √p) + (my_id + 1) mod √p;
5.        result := my_msg;
6.        msg := result;
7.        for i := 1 to √p − 1 do
8.            send msg to right;
9.            receive msg from left;
10.           result := result ∪ msg;
11.       endfor;

/* Communication along columns */
12.       up := (my_id − √p) mod p;
13.       down := (my_id + √p) mod p;
14.       msg := result;
15.       for i := 1 to √p − 1 do
16.           send msg to down;
17.           receive msg from up;
18.           result := result ∪ msg;
19.       endfor;
20.   end ALL_TO_ALL_BC_MESH
```

Algorithm 4.6 All-to-all broadcast on a square mesh of p nodes.

Basic Communication Operations

(All-to-All Broadcast on 2D Mesh..Algorithm 4.6)



(a) Initial data distribution

(b) Data distribution after rowwise broadcast

Figure 4.10 All-to-all broadcast on a 3×3 mesh. The groups of nodes communicating with each other in each phase are enclosed by dotted boundaries. By the end of the second phase, all nodes get (0,1,2,3,4,5,6,7) (that is, a message from each node).

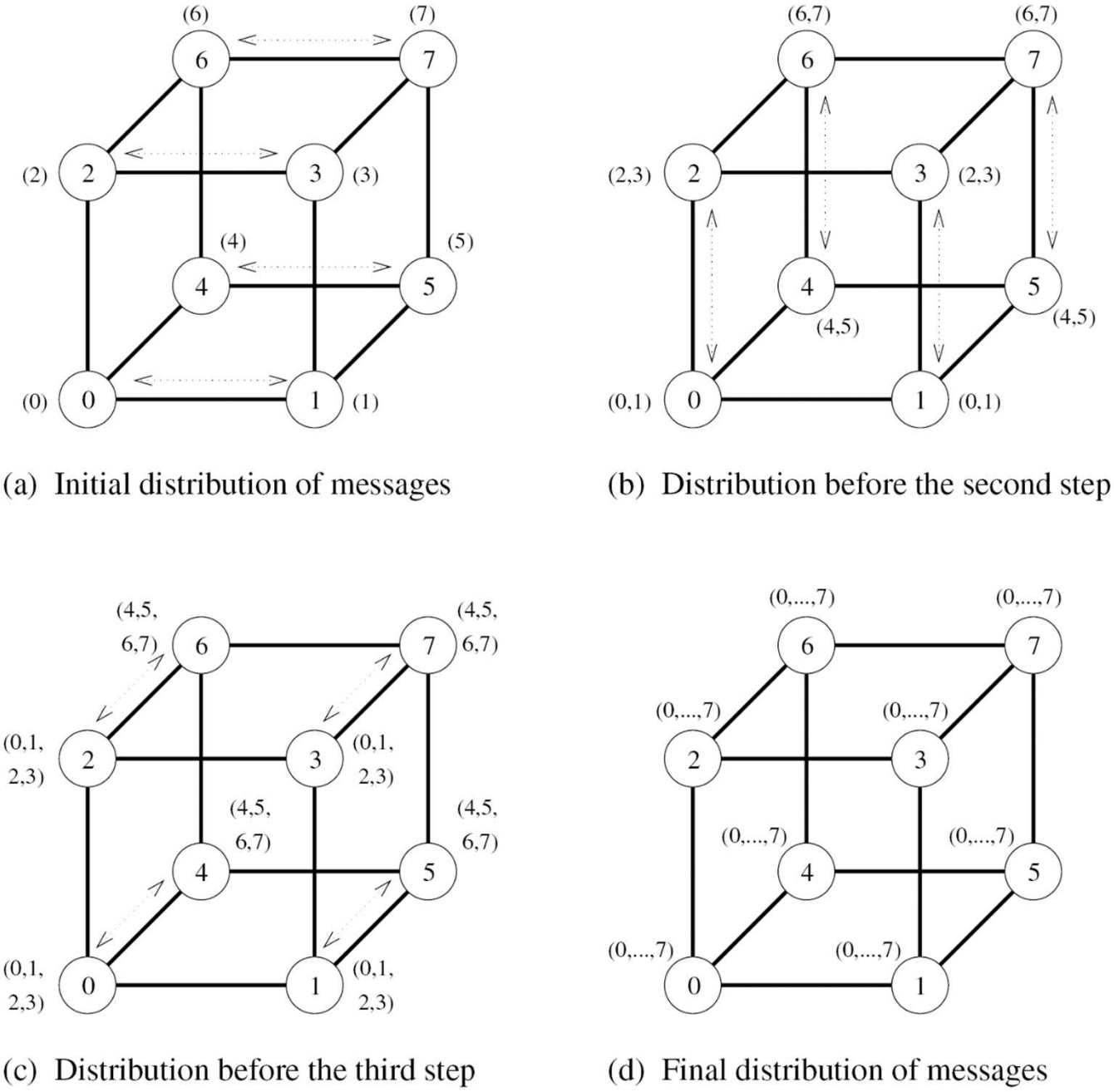


Figure 4.11 All-to-all broadcast on an eight-node hypercube.

Basic Communication Operations

(All-to-All Broadcast on HyperCube..Algorithm 4.6)

```
1.   procedure ALL_TO_ALL_BC_HCUBE(my_id, my_msg, d, result)
2.   begin
3.       result := my_msg;
4.       for i := 0 to d − 1 do
5.           partner := my_id XOR 2^i;
6.           send result to partner;
7.           receive msg from partner;
8.           result := result ∪ msg;
9.       endfor;
10.  end ALL_TO_ALL_BC_HCUBE
```

Algorithm 4.7 All-to-all broadcast on a d-dimensional hypercube.

Basic Communication Operations

(All-to-All Broadcast on HyperCube..Algorithm 4.6)

- Same algorithm for linear ring is not possible as in one-to-all broadcast

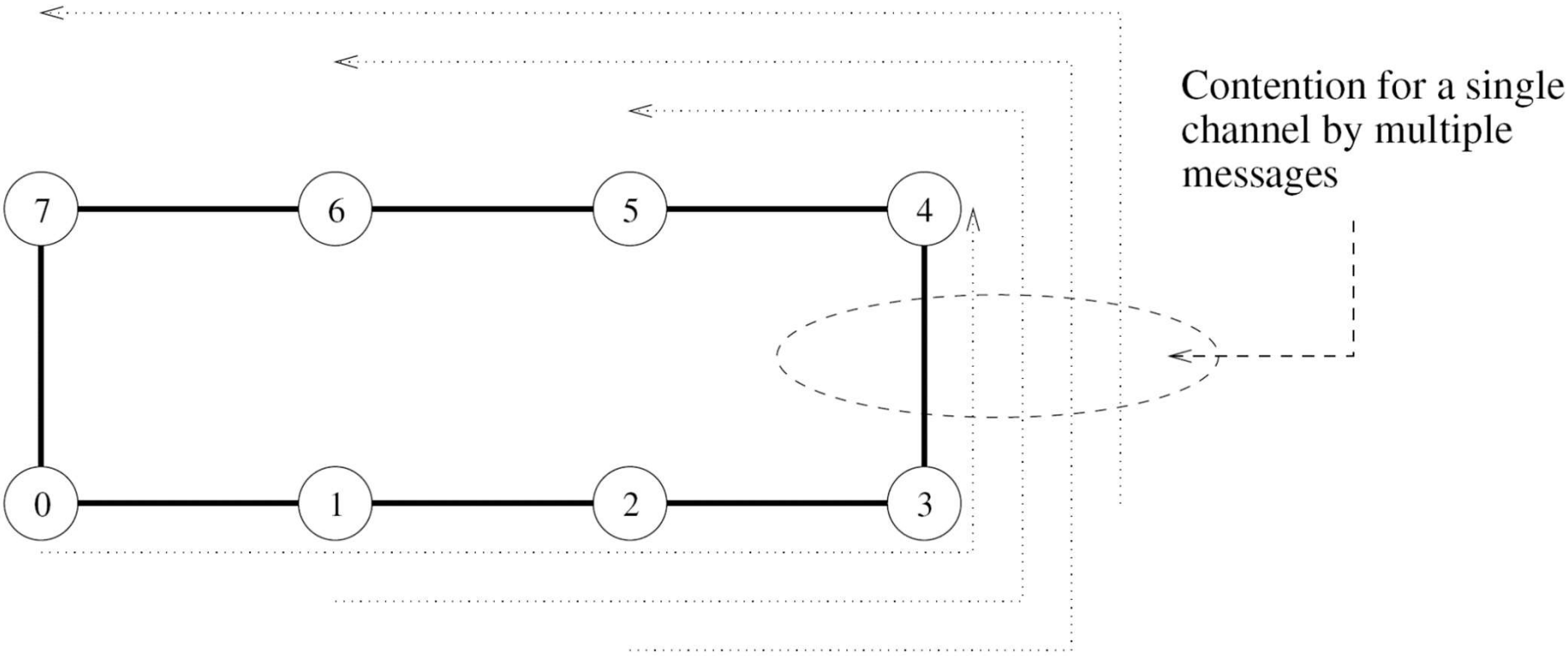


Figure 4.12 Contention for a channel when the communication step of Figure 4.11(c) for the hypercube is mapped onto a ring.

Basic Communication Operations

(All-to-All Broadcast and All-to-All Reduction)

Cost Estimation

- Different on each infrastructure.
- **Linear Ring**
  - $T = (t_s + mt_w)\,(p-1)$
- **Mesh**
  - Total time for All-to-All broadcast in the first phase
    - $T(first\ phase) = (t_s + mt_w)(\sqrt{p}-1)$
  - Total time for the second phase (note here m= $\sqrt{p}$.m)
    - $T(Second\ phase) = (t_s + (\sqrt{p})mt_w)(\sqrt{p}-1)$
- So, Total time= $2t_s(\sqrt{p}-1) + mt_w(p-1)$

Basic Communication Operations

(All-to-All Broadcast and All-to-All Reduction)

Cost Estimation

- Different on each infrastructure.
- **Hypercube (broadcast)**
  - Communication in for 1st step: $(\boldsymbol{t_s+mt_w})$
  - Communication in for 2nd step: $(\boldsymbol{t_s+2mt_w})$
  - Communication in for ith step: $(\boldsymbol{t_s+2^{i-1}mt_w})$
  - Total Cost= $\sum_{i=1}^{\log(p)}(t_s+2^{i-1}mt_w)$

Basic Communication Operations

(All-to-All Broadcast and All-to-All Reduction)

Cost Estimation

- Total Cost= $\sum_{i=1}^{\log(p)}(t_s+2^{i-1}mt_w)$
- Simplify the equation
- HINT: $[x^0 + x^1 + \ldots\ldots + x^n = \frac{x^{n+1}-1}{x-1}]$

- Answer $\boldsymbol{T = (t_s \log p + mt_w(p-1))}$

Questions

References

1. Kumar, V., Grama, A., Gupta, A., & Karypis, G. (1994). *Introduction to parallel computing* (Vol. 110). Redwood City, CA: Benjamin/Cummings.
2. Quinn, M. J. Parallel Programming in C with MPI and OpenMP,(2003).