Parallel and Distributed Computing

CS3006

Lecture 13

Basic Communication Operations-III

25th April 2022

Dr. Rana Asif Rehman

All-Reduce

Basic Communication Operations **(All-Reduce)**

- Precondition: Every process *i* has a single message M_i of size *m words.*
- Post condition: All processes have a reduced message *M of size m words.*

Strategies:

1. Use **all-to-one reduction** followed by **one-to-all** broadcast ($\mathbf{2} * (\boldsymbol{t_s} + \boldsymbol{m}\boldsymbol{t_w}) \log \boldsymbol{p}$)
2. Use **modified All-to-All comm.** algorithm for hypercube ($(\boldsymbol{t_s} + \boldsymbol{m}\boldsymbol{t_w}) \log \boldsymbol{p}$)
   - Replace Union with associative operator

Prefix-Sum

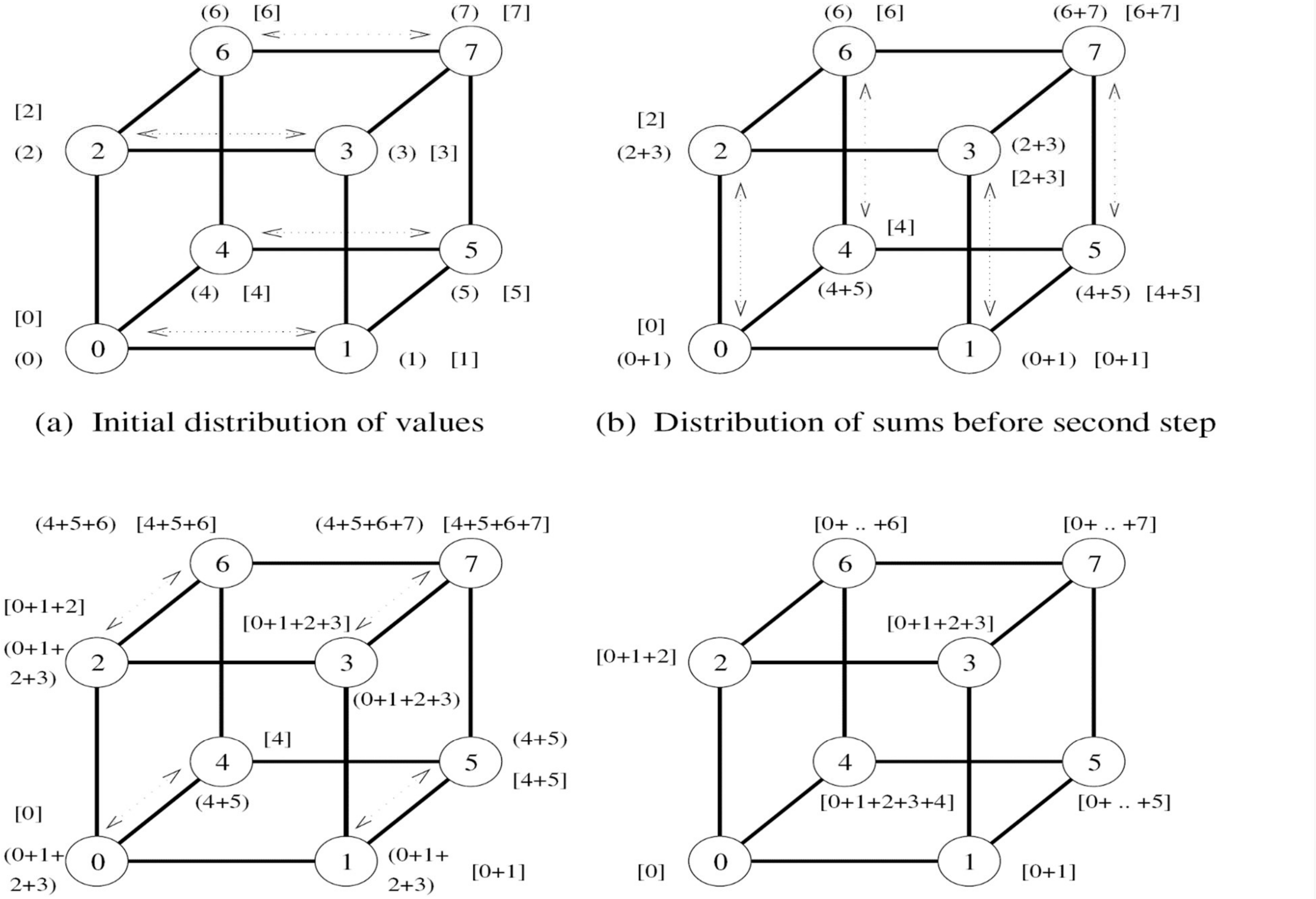
Basic Communication Operations **(Prefix-Sums)**

- Prefix-sums are also known as scan operations
- Given ***p*** numbers ***n*₀, *n*₁, …, *n*ₚ₋₁** (one on each node), the problem is to compute the sums such that: -
  - $\boldsymbol{S_k = \sum_{i=0}^{K}(n_i)}$
    - Here $\boldsymbol{S_k}$ is the prefix-sum computed at kth node after the operation.

Example:

- Original sequence: <3, 1, 4, 0, 2>
- Sequence of prefix sums: <3, 4, 8, 8, 10>

Basic Communication Operations **(Prefix-Sums)**



(a) Initial distribution of values

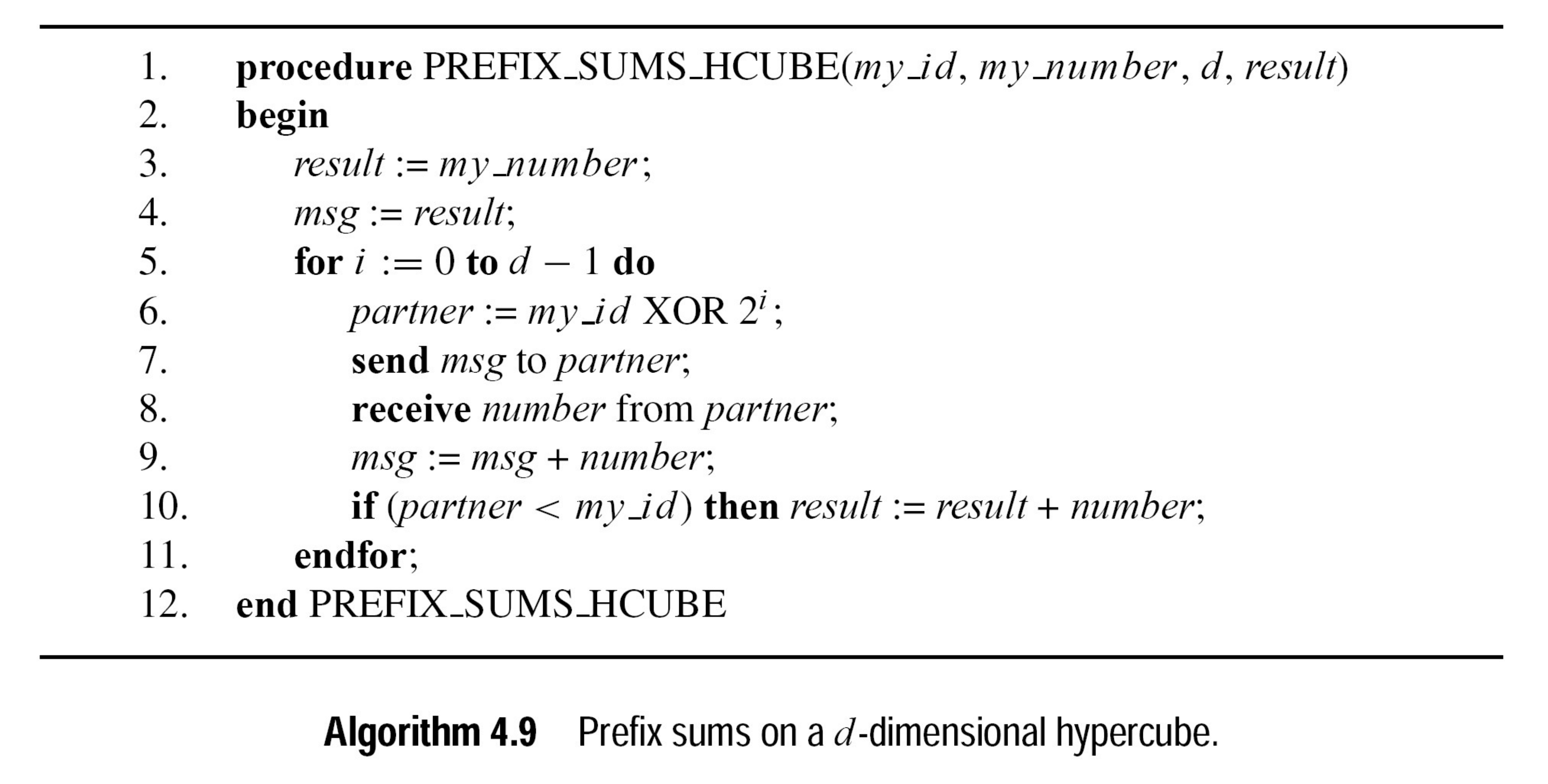
(b) Distribution of sums before second step

(c) Distribution of sums before third step

(d) Final distribution of prefix sums

Figure 4.13 Computing prefix sums on an eight-node hypercube. At each node, square brackets

Basic Communication Operations **(Prefix-Sums)**



```
procedure PREFIX_SUMS_HCUBE($my\_id$, $my\_number$, $d$, $result$)
begin
    $result := my\_number$;
    $msg := result$;
    for $i := 0$ to $d - 1$ do
        $partner := my\_id$ XOR $2^i$;
        send $msg$ to $partner$;
        receive $number$ from $partner$;
        $msg := msg + number$;
        if ($partner < my\_id$) then $result := result + number$;
    endfor;
end PREFIX_SUMS_HCUBE
```

Algorithm 4.9 Prefix sums on a d-dimensional hypercube.

Scatter and Gather

Basic Communication Operations **(Scatter and Gather)**

- Gather is different than reduction as it doesn’t reduce the results with associative operator

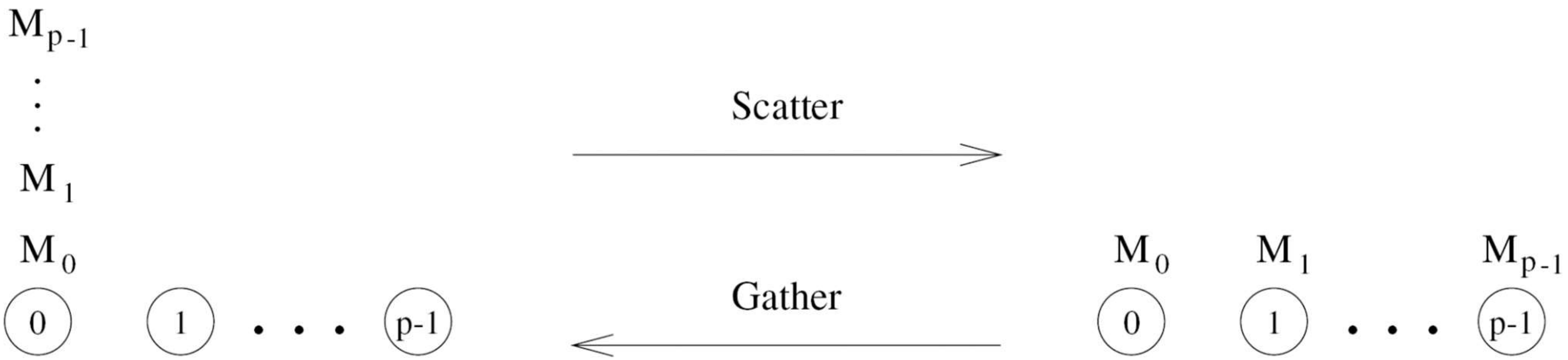
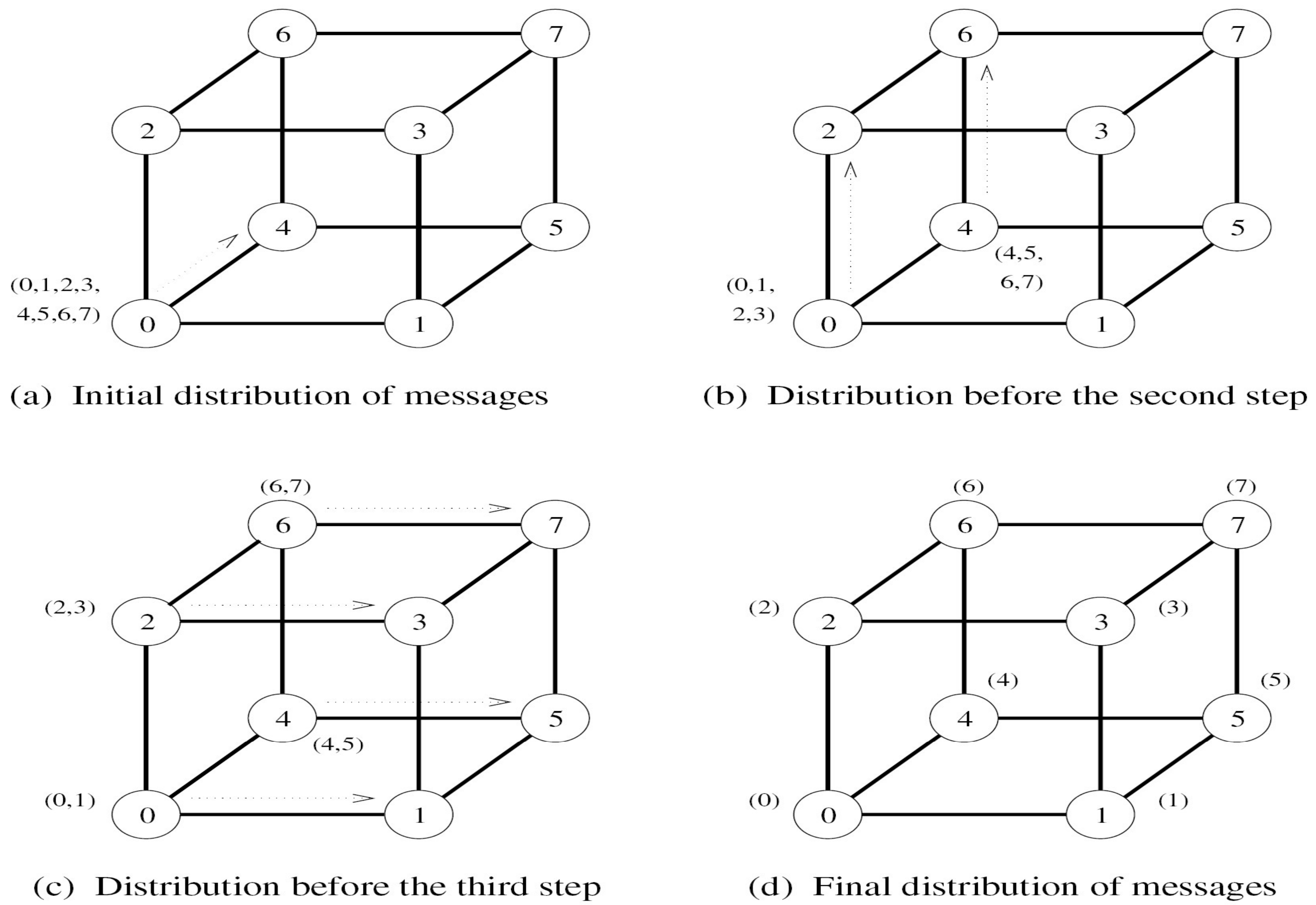


Figure 4.14 Scatter and gather operations.

Basic Communication Operations



(a) Initial distribution of messages

(b) Distribution before the second step

(c) Distribution before the third step

(d) Final distribution of messages

Figure 4.15 The scatter operation on an eight-node hypercube.

All-to-All personalized Communication

Basic Communication Operations

(All-to-All personalized)

- Each node sends a distinct message of size ***m*** to every other node.
- Also known **total exchange**

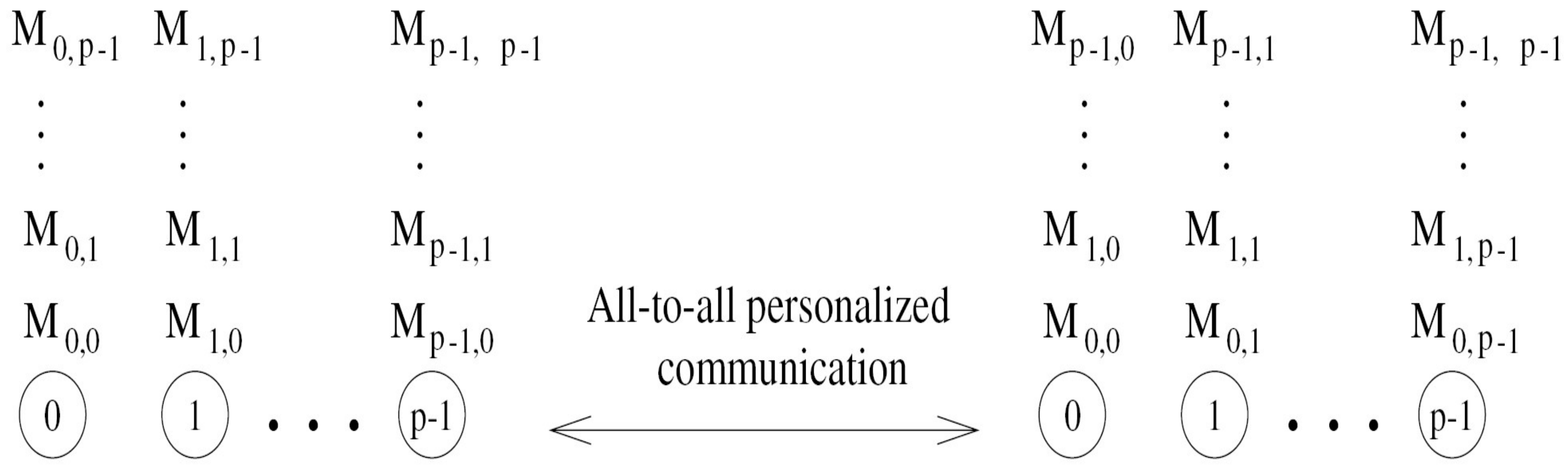
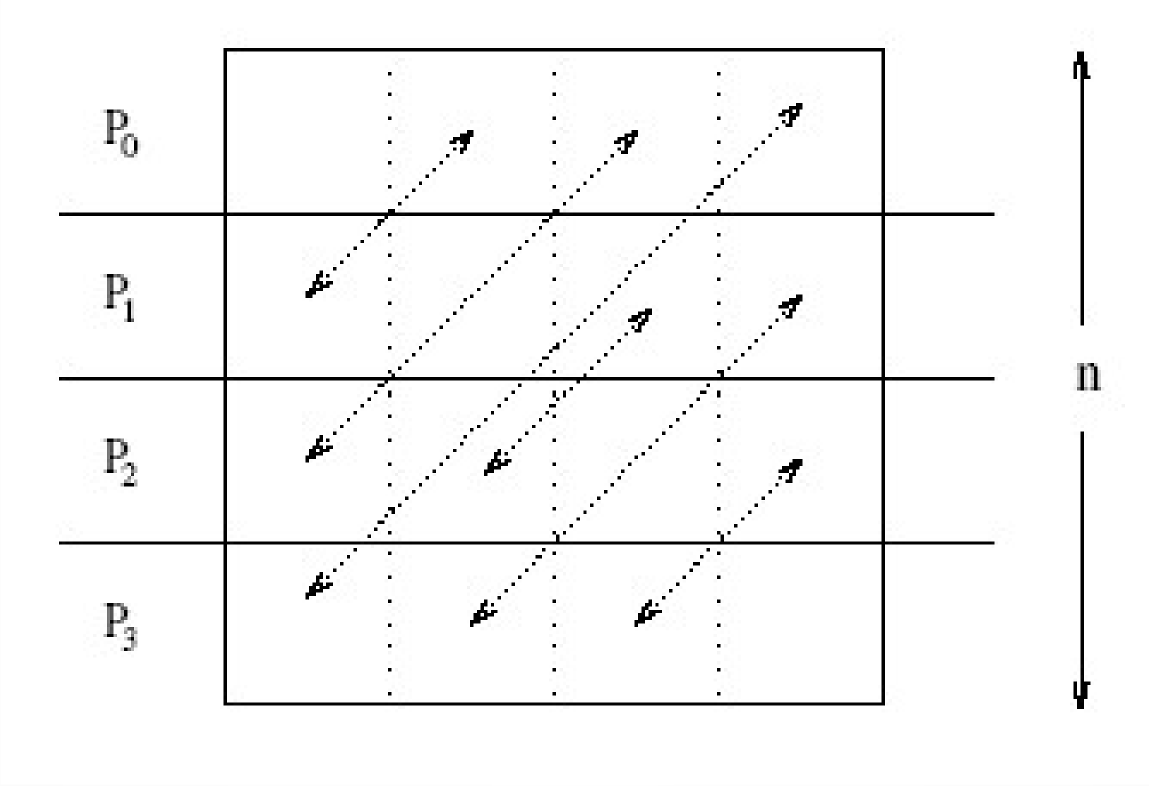


Figure 4.16 All-to-all personalized communication.

Basic Communication Operations

(All-to-All personalized)



All-to-all personalized communication in transposing a *4 x 4* matrix using four processes.

Basic Communication Operations

(All-to-All personalized [Ring])

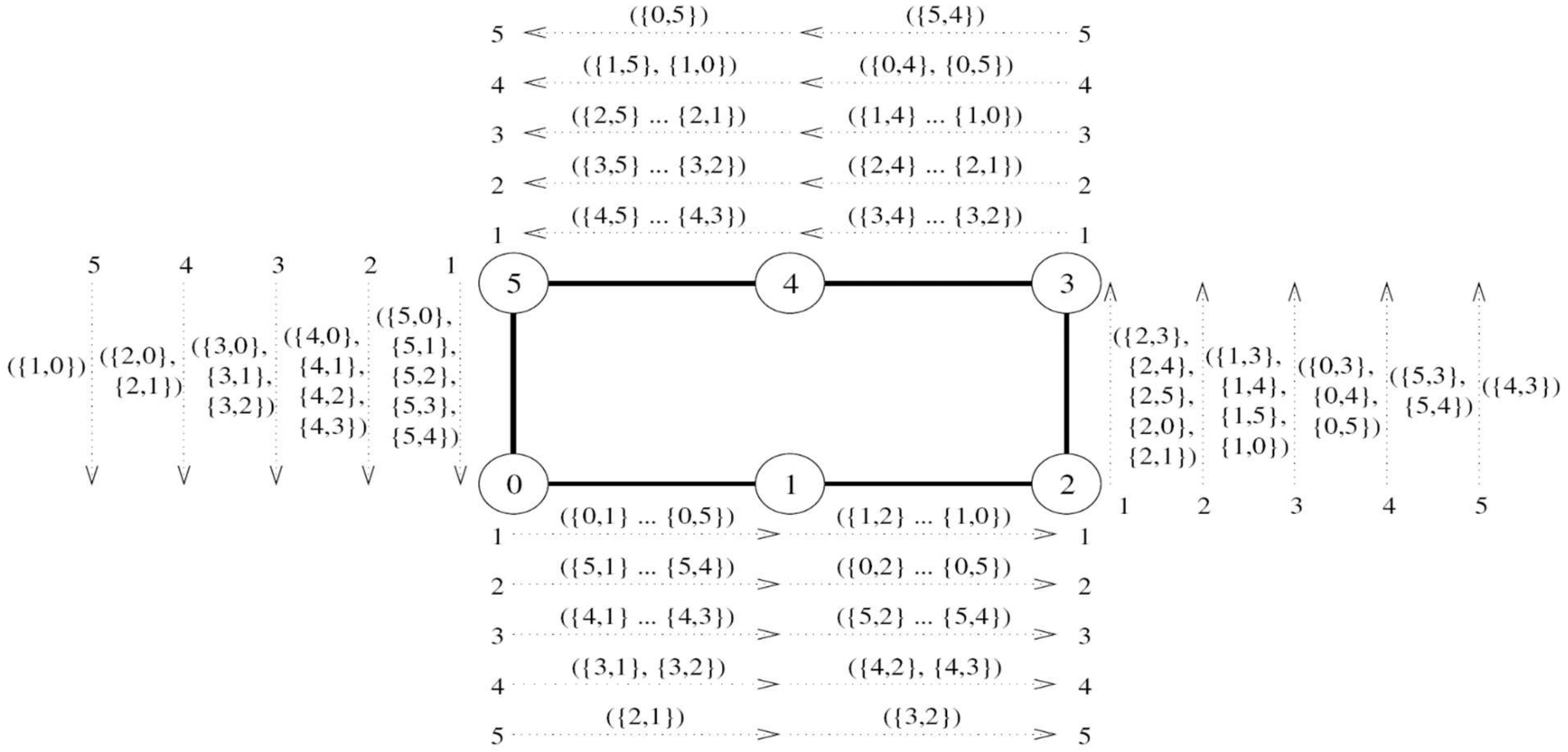


Figure 4.18 All-to-all personalized communication on a six-node ring. The label of each message is of the form $\{x, y\}$, where x is the label of the node that originally owned the message, and y is the label of the node that is the final destination of the message. The label $(\{x_1, y_1\}, \{x_2, y_2\}, \ldots, \{x_n, y_n\})$ indicates a message that is formed by concatenating n individual messages.

Basic Communication Operations

(All-to-All personalized [Ring])

Cost Analysis

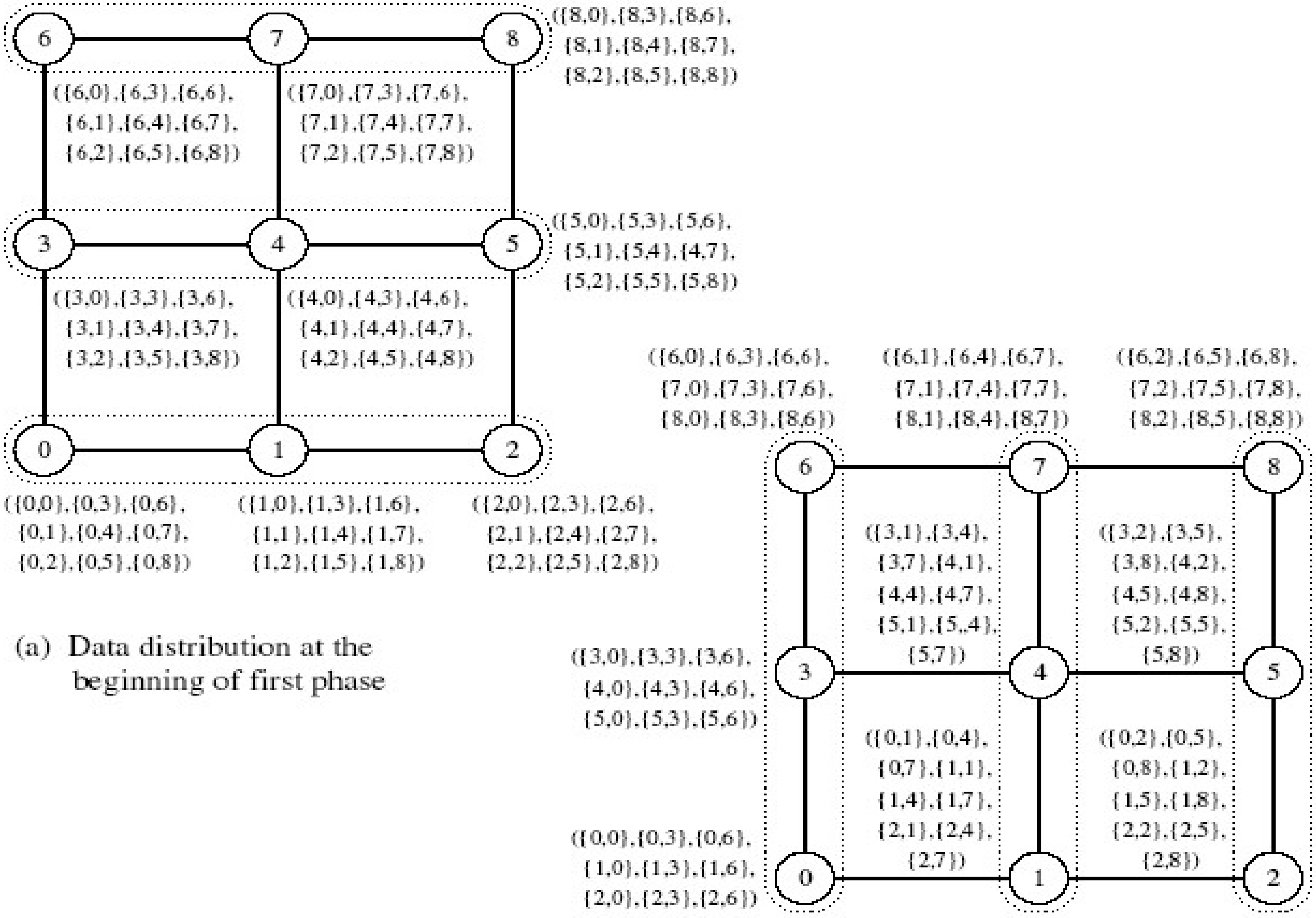
- $T = \sum_{i=1}^{(p-1)}(t_s + (p-i)mt_w)$
  - $= \sum_{i=1}^{(p-1)}(t_s) + \ mt_w \sum_{i=1}^{(p-1)}(p-i)\ )$

  → $(\mathbf{p-1})(t_s) \quad + \quad mt_w \sum_{i=1}^{(p-1)}(\boldsymbol{i})$

  → $\left((t_s + \left(\frac{1}{2}\right)\boldsymbol{pm}t_w\right)(p-1)$

Basic Communication Operations

(All-to-All personalized [Mesh])



(a) Data distribution at the beginning of first phase

(b) Data distribution at the beginning of second phase

Basic Communication Operations

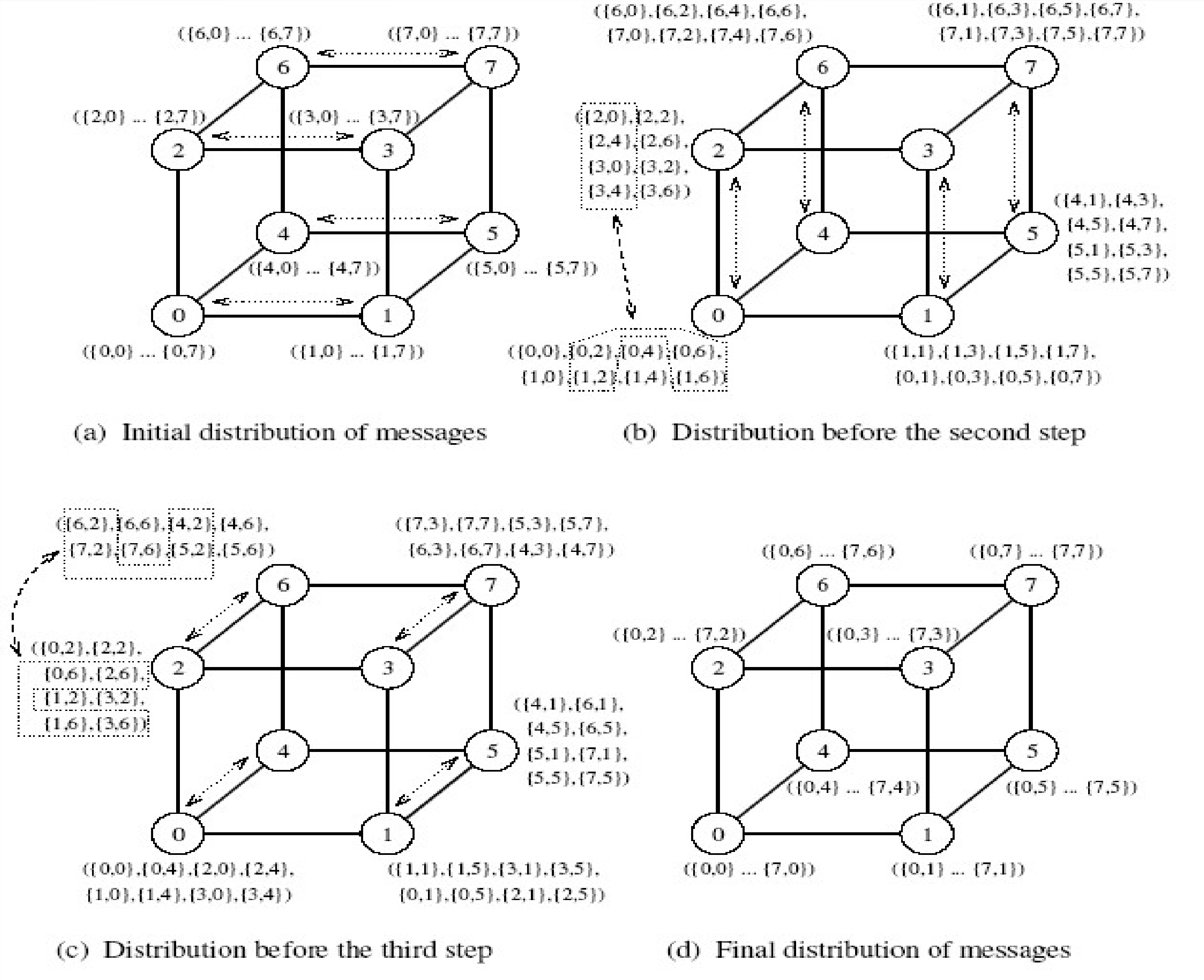
(All-to-All personalized [Mesh])

Cost Analysis

- Time for the first phase is identical to that in a ring with √*p* processors, i.e., *(t_s + $t_w$mp/2)(√p – 1)*.
  - Here **m*t*$_w$** *becomes* **√*p* m*t*$_w$** and **P** becomes **√*p***
- Time in the second phase is identical to the first phase. Therefore, total time is twice of this time, i.e.,

Basic Communication Operations

(All-to-All personalized [Hyper Cube])



(a) Initial distribution of messages

(b) Distribution before the second step

(c) Distribution before the third step

(d) Final distribution of messages

Questions

References

1. Kumar, V., Grama, A., Gupta, A., & Karypis, G. (2017). *Introduction to parallel computing*. Redwood City, CA: Benjamin/Cummings.