



Parallel and Distributed Computing

CS3006

Lecture 11

Basic Communication Operations

18th March 2024

Dr. Rana Asif Rehman



Basic Communication Operations

Parallel and Distributed Computing (CS3006) - Spring 2024

Basic Communication Operations

Preliminaries

- Exchanging the data is fundamental requirement for most of the parallel algorithms
- $t_s + mt_w$ - the simplified communication cost model :-
 - Over distributed memory infrastructure
 - Assuming the cut-through routing
- The chapter is about commonly used basic communication patterns over the different interconnections
 - We shall drive communication costs of these operations on different interconnections.

Cut-through routing is a packet-switching method used by network devices such as switches and routers to increase their speed¹². It works by forwarding a data packet as soon as it has read its destination address, without waiting for the entire packet to be received

Basic Communication Operations

Assumptions for the Operations

- Interconnections support cut-through routing
- Communication time between any pair of nodes in the network is same (regardless of the number of intermediate nodes)
- Links are bi-directional
 - The directly connected nodes can simultaneously send messages of m words without any congestion
- Single-port communication model
 - A node can send on only one of its links at a time
 - A node can receive on only one of its links at a time
- However, a node can receive a message while sending another message at the same time on the same or a different link.
- Consider $p=2^i$ nodes



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

Basic Communication Operations

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

One-to-All Broadcast

- A single process sends identical data to all other processes.
 - Initially one process has data of m size.
 - After broadcast operation, each of the processes have own copy of the m size.

All-to-One Reduction

- Dual of one-to-all broadcast
- The m -sized data from all processes are combined through an associative operator
- accumulated at a single destination process into one buffer of size m

Basic Communication Operations

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

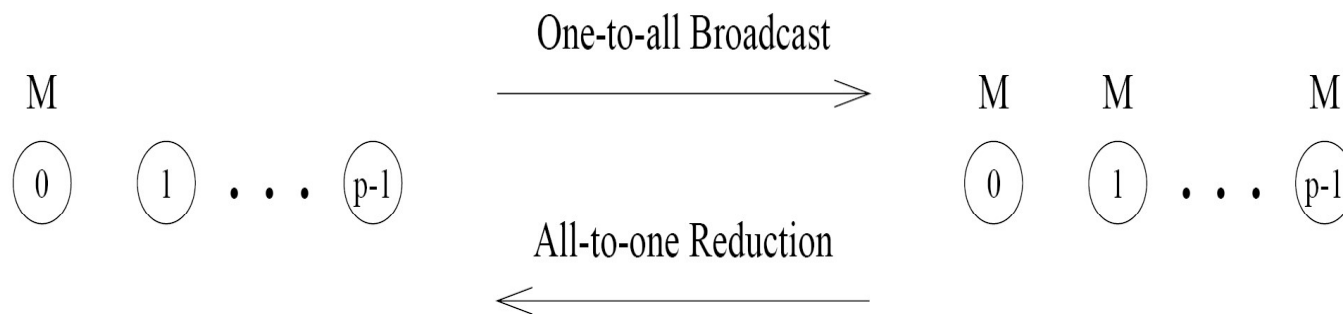


Figure 4.1 One-to-all broadcast and all-to-one reduction.

Basic Communication Operations

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

Linear Array or Ring

➤ Naïve solution

- sequentially send $p - 1$ messages from the source to the other $p - 1$ processes
 - Bottle necks, and underutilization of communication network
- Solution?

➤ Recursive doubling

- Source process sends the message to another process
- In next communication phase both the processes can simultaneously propagate the message

Basic Communication Operations

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

Linear Array or Ring

Recursive Doubling Broadcast

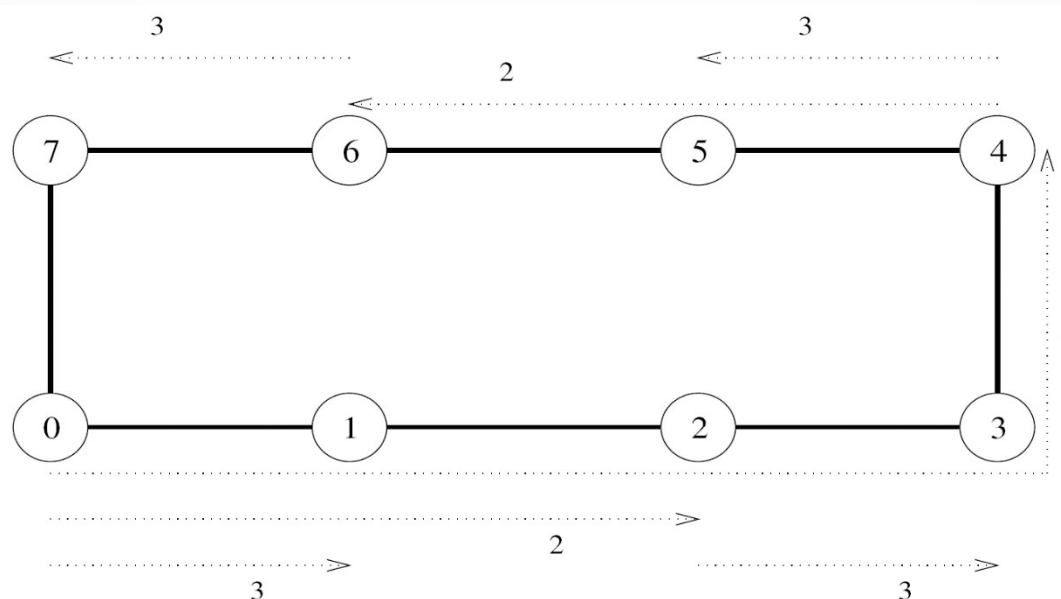


Figure 4.2 One-to-all broadcast on an eight-node ring. Node 0 is the source of the broadcast. Each message transfer step is shown by a numbered, dotted arrow from the source of the message to its destination. The number on an arrow indicates the time step during which the message is transferred.

Basic Communication Operations

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

Linear Array or Ring

➡ Recursive Doubling Reduction

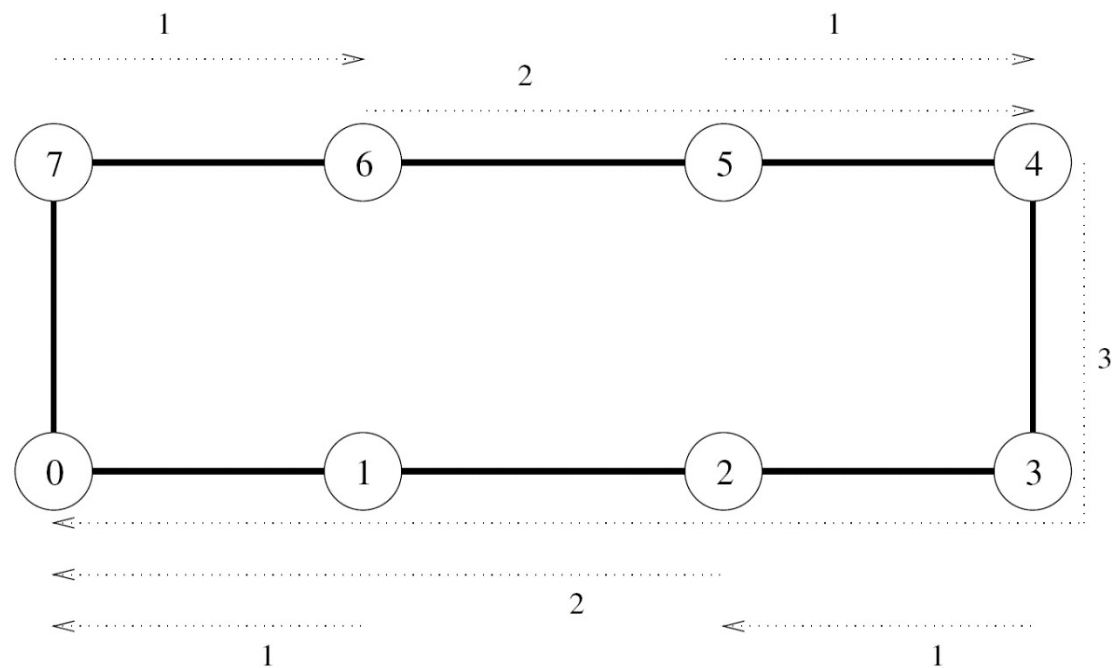


Figure 4.3 Reduction on an eight-node ring with node 0 as the destination of the reduction.

Basic Communication Operations

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

Matrix-Vector Multiplication (An Application)

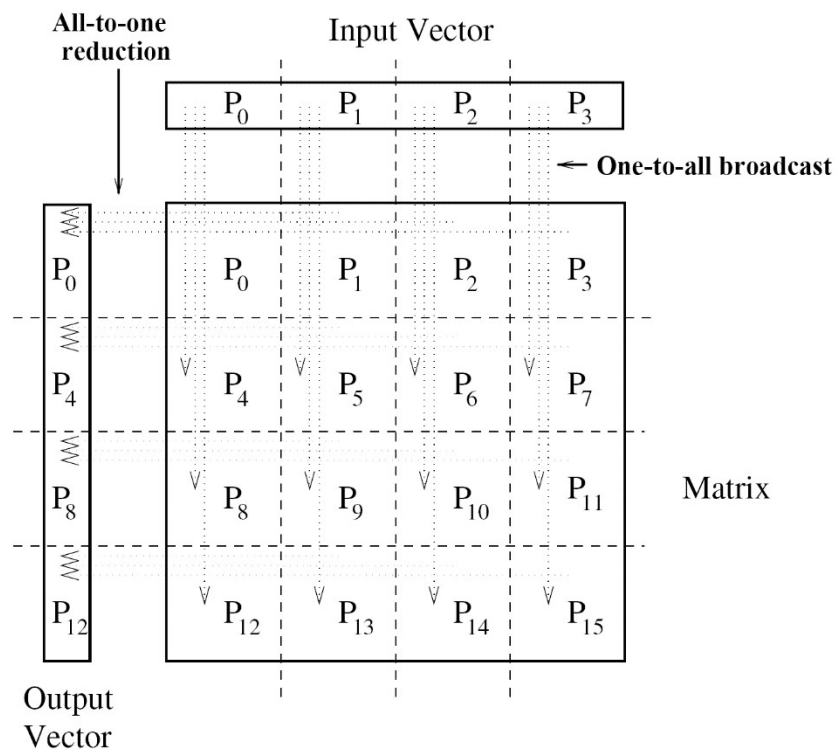


Figure 4.4 One-to-all broadcast and all-to-one reduction in the multiplication of a 4×4 matrix with a 4×1 vector.

Basic Communication Operations

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

Mesh

- We can regard each row and column of a square mesh of p nodes as a linear array of nodes
- Communication algorithms on the mesh are simple extensions of their linear array counterparts

➤ Broadcast and Reduction

- Two step breakdown:
 - I. The operation is performed along one by treating the row as linear array
 - II. Then the all the columns are treated similarly

Basic Communication Operations

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

Mesh (Broadcast and Reduction)

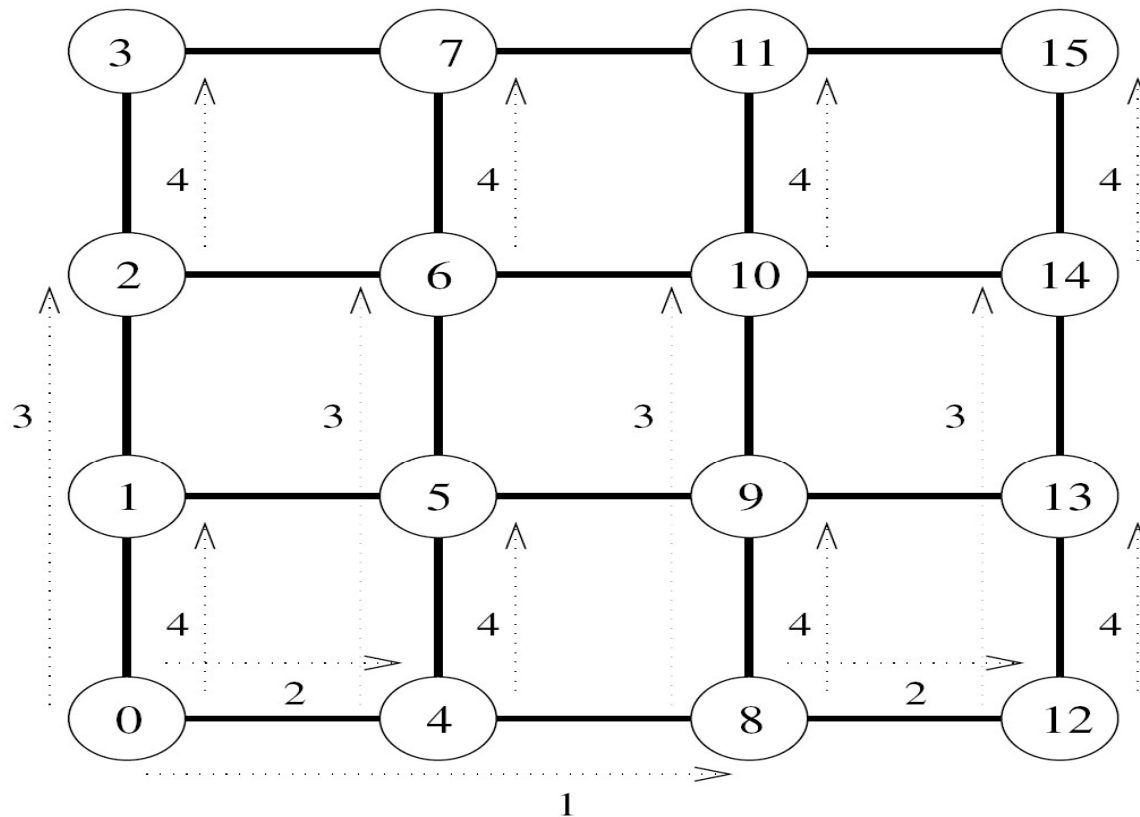


Figure 4.5 One-to-all broadcast on a 16-node mesh.

Basic Communication Operations

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

Balanced Binary Tree

➡ Broadcast

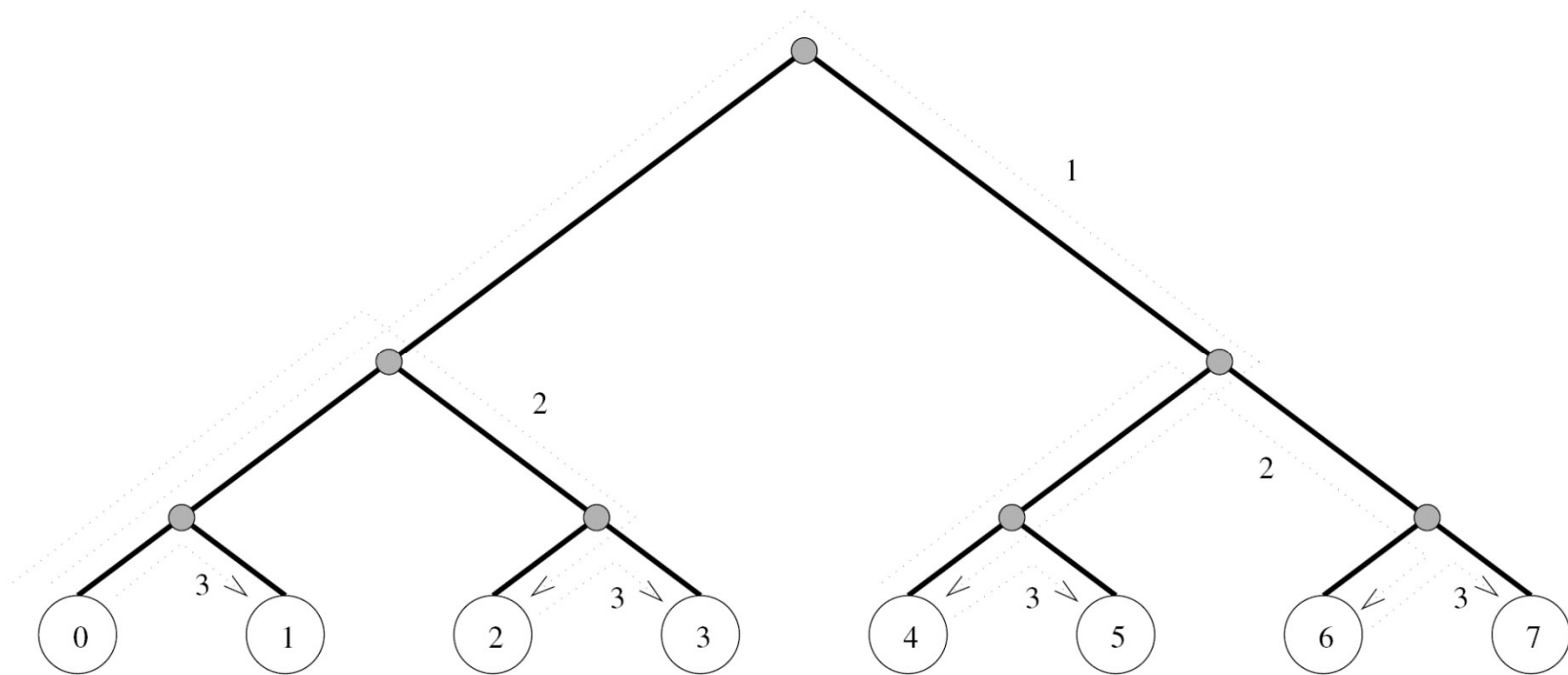


Figure 4.7 One-to-all broadcast on an eight-node tree.

Basic Communication Operations

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

Hypercube

Broadcast

- Source node first send data to one node in the highest dimension
- The communication successively proceeds along lower dimensions in the subsequent steps
- The algorithm is same as used for linear array
 - But, here changing order of dimension does not congest the network

Basic Communication Operations

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

Hypercube

➤ Broadcast

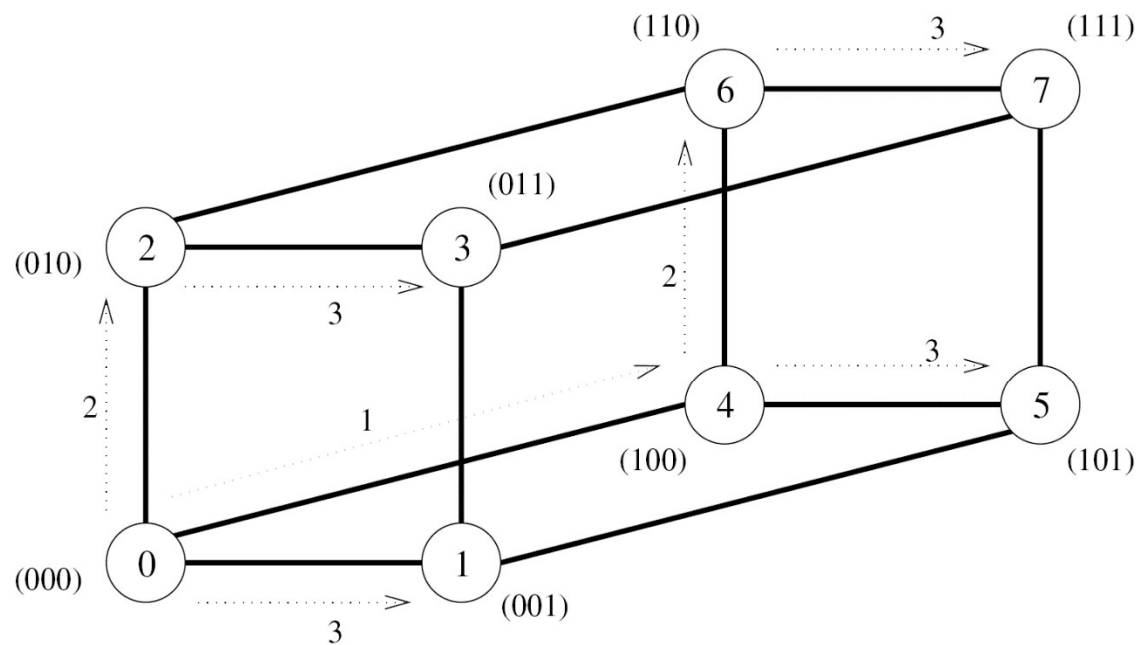


Figure 4.6 One-to-all broadcast on a three-dimensional hypercube. The binary representations of node labels are shown in parentheses.

Basic Communication Operations

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

```
1.  procedure ONE_TO_ALL_BC( $d, my\_id, X$ )
2.  begin
3.       $mask := 2^d - 1;$                                 /* Set all  $d$  bits of  $mask$  to 1 */
4.      for  $i := d - 1$  downto 0 do                        /* Outer loop */
5.           $mask := mask \text{ XOR } 2^i;$                     /* Set bit  $i$  of  $mask$  to 0 */
6.          if ( $my\_id \text{ AND } mask$ ) = 0 then /* If lower  $i$  bits of  $my\_id$  are 0 */
7.              if ( $my\_id \text{ AND } 2^i$ ) = 0 then
8.                   $msg\_destination := my\_id \text{ XOR } 2^i;$ 
9.                  send  $X$  to  $msg\_destination$ ;
10.             else
11.                  $msg\_source := my\_id \text{ XOR } 2^i;$ 
12.                 receive  $X$  from  $msg\_source$ ;
13.             endelse;
14.          endif;
15.      endfor;
16.  end ONE_TO_ALL_BC
```

Algorithm 4.1 One-to-all broadcast of a message X from node 0 of a d -dimensional p -node hypercube ($d = \log p$). AND and XOR are bitwise logical-and and exclusive-or operations, respectively.

Basic Communication Operations

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

```
1.  procedure GENERAL_ONE_TO_ALL_BC( $d, my\_id, source, X$ )
2.  begin
3.     $my\_virtual\_id := my\_id \text{ XOR } source$ ;
4.     $mask := 2^d - 1$ ;
5.    for  $i := d - 1$  downto 0 do    /* Outer loop */
6.       $mask := mask \text{ XOR } 2^i$ ; /* Set bit  $i$  of  $mask$  to 0 */
7.      if ( $my\_virtual\_id \text{ AND } mask$ ) = 0 then
8.        if ( $my\_virtual\_id \text{ AND } 2^i$ ) = 0 then
9.           $virtual\_dest := my\_virtual\_id \text{ XOR } 2^i$ ;
10.         send  $X$  to ( $virtual\_dest \text{ XOR } source$ );
11.        /* Convert  $virtual\_dest$  to the label of the physical destination */
12.      else
13.         $virtual\_source := my\_virtual\_id \text{ XOR } 2^i$ ;
14.        receive  $X$  from ( $virtual\_source \text{ XOR } source$ );
15.        /* Convert  $virtual\_source$  to the label of the physical source */
16.      endelse;
17.    endfor;
18.  end GENERAL_ONE_TO_ALL_BC
```

Algorithm 4.2 One-to-all broadcast of a message X initiated by $source$ on a d -dimensional hypothetical hypercube. The AND and XOR operations are bitwise logical operations.

Basic Communication Operations

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

```
1.  procedure ALL_TO_ONE_REDUCE( $d, my\_id, m, X, sum$ )
2.  begin
3.      for  $j := 0$  to  $m - 1$  do  $sum[j] := X[j];$ 
4.       $mask := 0;$ 
5.      for  $i := 0$  to  $d - 1$  do
6.          /* Select nodes whose lower  $i$  bits are 0 */
7.          if  $(my\_id \text{ AND } mask) = 0$  then
8.              if  $(my\_id \text{ AND } 2^i) \neq 0$  then
9.                   $msg\_destination := my\_id \text{ XOR } 2^i;$ 
10.                 send  $sum$  to  $msg\_destination;$ 
11.             else
12.                  $msg\_source := my\_id \text{ XOR } 2^i;$ 
13.                 receive  $X$  from  $msg\_source;$ 
14.                 for  $j := 0$  to  $m - 1$  do
15.                      $sum[j] := sum[j] + X[j];$ 
16.                 endelse;
17.                  $mask := mask \text{ XOR } 2^i; \quad /* \text{Set bit } i \text{ of } mask \text{ to } 1 */$ 
18.             endfor;
19.  end ALL_TO_ONE_REDUCE
```

Basic Communication Operations

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

Cost Estimation

- Broadcast needs **$\log(p)$** point-to-point simple message transfer steps.
- Message size of each transfer is **m**
- Time for each of the transfers is: **$t_s + mt_w$**

Hence cost for $\log(p)$ transfers = $T = (t_s + mt_w) \log p$

Questions



Parallel and Distributed Computing
(CS3006) - Spring 2024



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).