procedure ONE_TO_ALL_BC(d, my_id, X) begin
    mask := 2^d − 1; /* Set all d bits of mask to 1 */
    for i := d − 1 downto 0 do /* Outer loop */
        mask := mask XOR 2^i; /* Set bit i of mask to 0 */
        if (my_id AND mask) = 0 then /* If lower i bits of my_id are 0 */
            if (my_id AND 2^i) = 0 then
                msg_destination := my_id XOR 2^i;
                send X to msg_destination;
            else
                msg_source := my_id XOR 2^i;
                receive X from msg_source;
            endelse;
        endif;
    endfor;
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.
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.
Algorithm 4.3  Single-node accumulation on a $d$-dimensional hypercube. Each node contributes a message $X$ containing $m$ words, and node 0 is the destination of the sum. The AND and XOR operations are bitwise logical operations.
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.
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.
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 \(\sqrt{p}\)) + (my_id − 1) mod \(\sqrt{p}\);
4. right := my_id − (my_id mod \(\sqrt{p}\)) + (my_id + 1) mod \(\sqrt{p}\);
5. result := my_msg;
6. msg := result;
7. for i := 1 to \(\sqrt{p} − 1\) do
8. send msg to right;
9. receive msg from left;
10. result := result \(\cup\) msg;
11. endfor;

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

Algorithm 4.6 All-to-all broadcast on a square mesh of \(p\) nodes.
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.
Algorithm 4.8  All-to-all broadcast on a $d$-dimensional hypercube. AND and XOR are bitwise logical-and and exclusive-or operations, respectively.
1. \textbf{procedure} PREFIX\_SUMS\_HCUBE(id, number, d, result) \\
2. \textbf{begin} \\
3. result := number;  \\
4. msg := result; \\
5. for \( i := 0 \) to \( d - 1 \) do \\
6. partner := id XOR \( 2^i \); \\
7. send msg to partner; \\
8. receive number from partner; \\
9. msg := msg + number; \\
10. if (partner < id) then result := result + number; \\
11. endfor; \\
12. end PREFIX\_SUMS\_HCUBE \\

\textbf{Algorithm 4.9} Prefix sums on a \( d \)-dimensional hypercube.
Algorithm 4.10 A procedure to perform all-to-all personalized communication on a $d$-dimensional hypercube. The message $M_{i,j}$ initially resides on node $i$ and is destined for node $j$. 

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>procedure ALL_TO_ALL_PERSONAL($d, my_id$)</td>
</tr>
<tr>
<td>2</td>
<td>begin</td>
</tr>
<tr>
<td>3</td>
<td>for $i := 1$ to $2^d - 1$ do</td>
</tr>
<tr>
<td>4</td>
<td>begin</td>
</tr>
<tr>
<td>5</td>
<td>partner := $my_id$ XOR $i$;</td>
</tr>
<tr>
<td>6</td>
<td>send $M_{my_id, partner}$ to partner;</td>
</tr>
<tr>
<td>7</td>
<td>receive $M_{partner, my_id}$ from partner;</td>
</tr>
<tr>
<td>8</td>
<td>endfor;</td>
</tr>
<tr>
<td>9</td>
<td>end ALL_TO_ALL_PERSONAL</td>
</tr>
</tbody>
</table>