Electrical Engineering and Computer Science

EECS 358 - INTRODUCTION TO PARALLEL COMPUTING

Lecture 9

Dist. Memory Message Communication

Outline

- One-to-all Broadcast
- All-to-all Broadcast
- One-to-all Scatter
- All-to-all Scatter

One-to-all Broadcast

- Algorithms often require a processor to send identical data to all other processors or a subset of processors. This operation is called a one-to-all broadcast or singlenode broadcast

- At the start of a singlenode broadcast, a processor has m words of data that needs to be sent, at the end there are p copies of this data, one on each processor

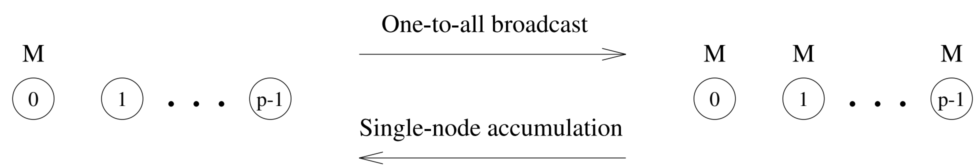
- The dual of a broadcast operation is a all-to-one reduction or singlenode reduction

- At the start of a singlenode reduction each processor has m words of data, the reduction combines all the data from processors using an associative operator to produce m words at the receiver

- Naive singlenode broadcast or reduction using $p - 1$ steps

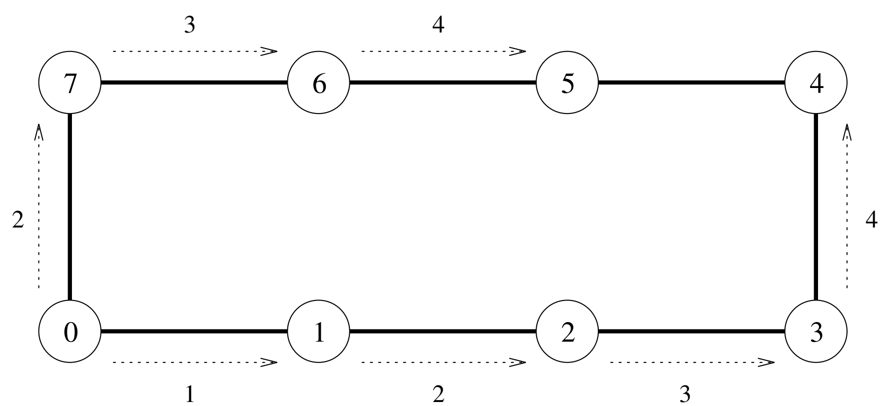
- Scatter is similar to broadcast, but each node gets a different data element

One-to-all Broadcast



Store-and-forward Routing on Ring

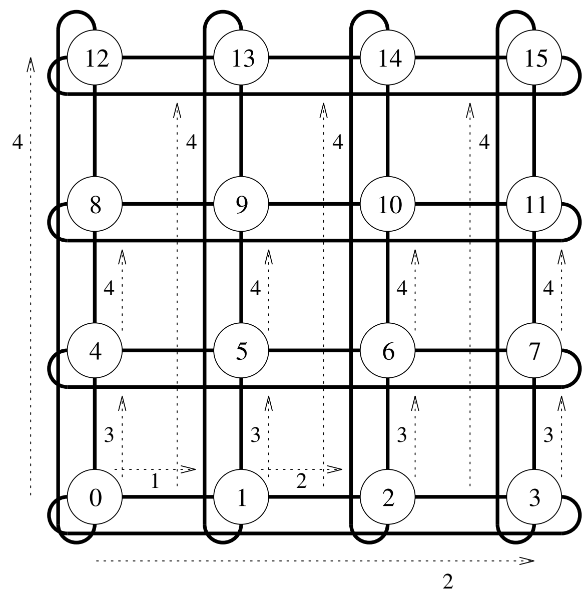
- Source sends message on both outgoing links in first two steps
- All other processors receive on a link and transmit on other link
- $\lceil \frac{p}{2} \rceil$ steps and $(t_s + t_w m)\lceil \frac{p}{2} \rceil$ cost



Store-and-forward Routing on 2d Torus

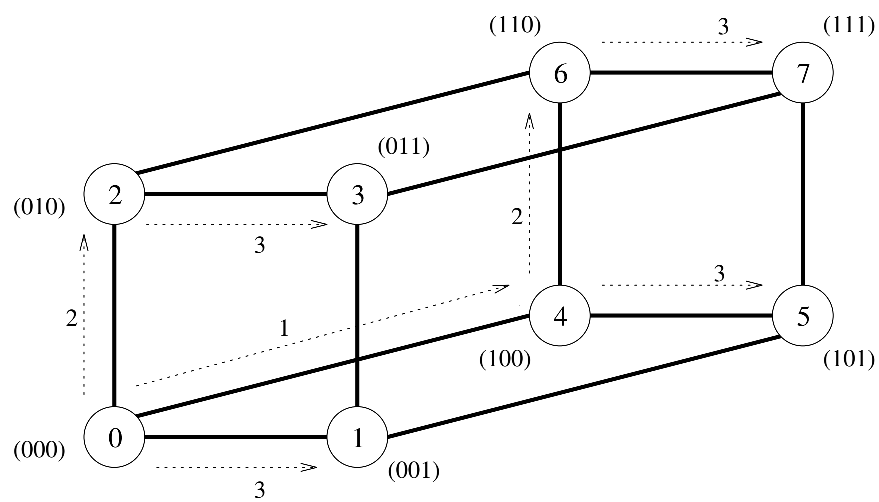
- Each row or column of the torus can be regarded as a ring: Use ring method for the row to which the sending processor belongs then use ring method for every column

- $2\lceil \frac{\sqrt{p}}{2} \rceil$ steps and $2(t_s + t_w m)\lceil \frac{\sqrt{p}}{2} \rceil$ cost



Store-and-forward Routing on Hypercube

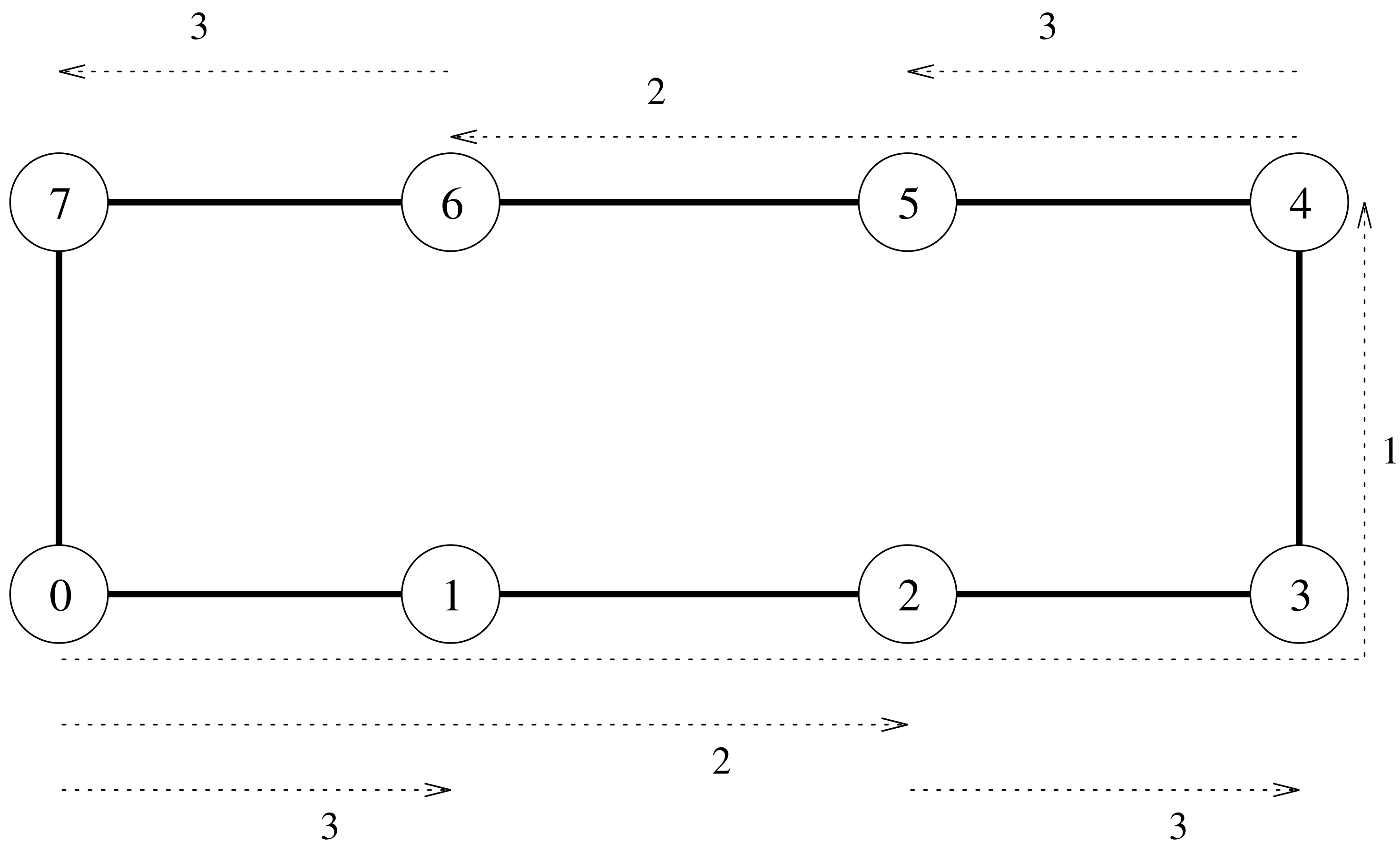
- Takes log(p) steps for a p processor hypercube
- In the ith step, all processors that have the message transmit it to the neighbouring processor that differs in the ith most significant bit
- $(t_s + t_w m)\log(p)$ cost



Cut-through Routing on Ring

- Algorithm similar to one used for hypercube; takes $\log(p)$ steps
- In step i, message is sent to processor at a distance $\frac{p}{2^i}$
- All messages flow in the same direction
- $t_s \log(p) + t_w m \log(p) + t_h(p-1)$ cost

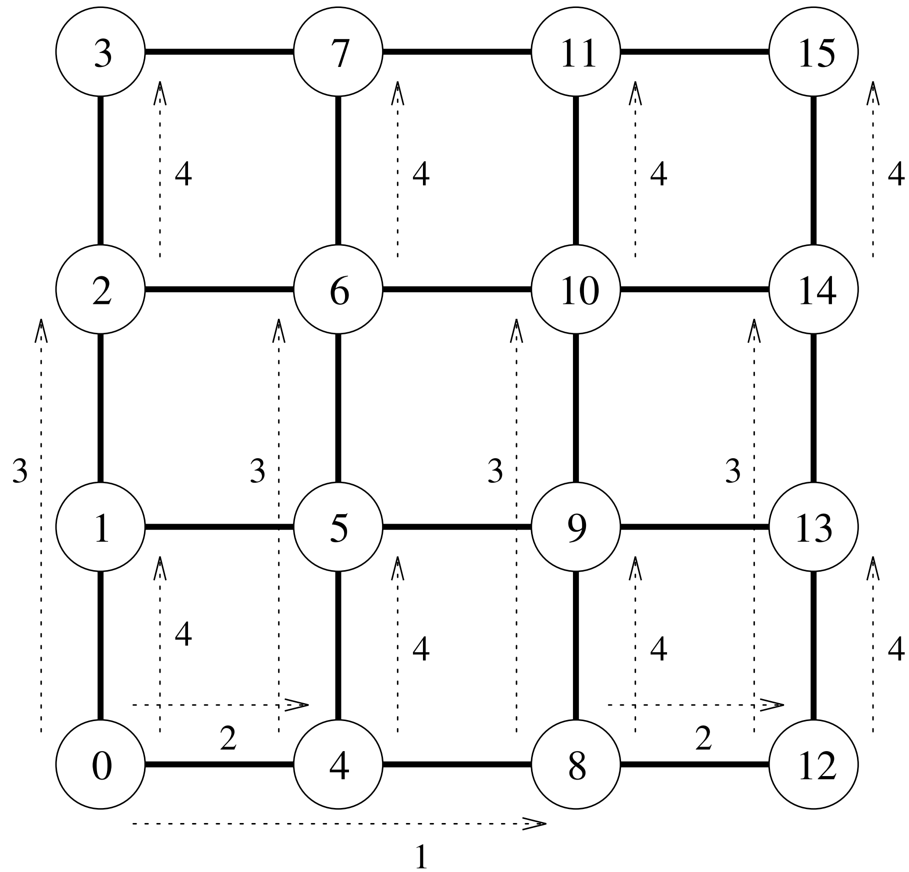
Cut-through Routing on Ring



Cut-through Routing on 2d Torus

- Apply ring algorithm for the processor row of sender
- Now use ring algorithm for all processor columns
- $2\log(\sqrt{p})$ steps
- $2(t_s + t_w m)\log(\sqrt{p}) + 2t_h(\sqrt{p} - 1)$ cost

Cut-through Routing on 2d Torus



Cut-through Routing on Hypercube

- For hypercube, cut-through does not provide benefits because of the use of only single link communications

All-to-all Broadcast

- Algorithms often require each processor to send identical data to all other processors or a subset of processors. This operation is called a all-to-all broadcast or a multinode broadcast

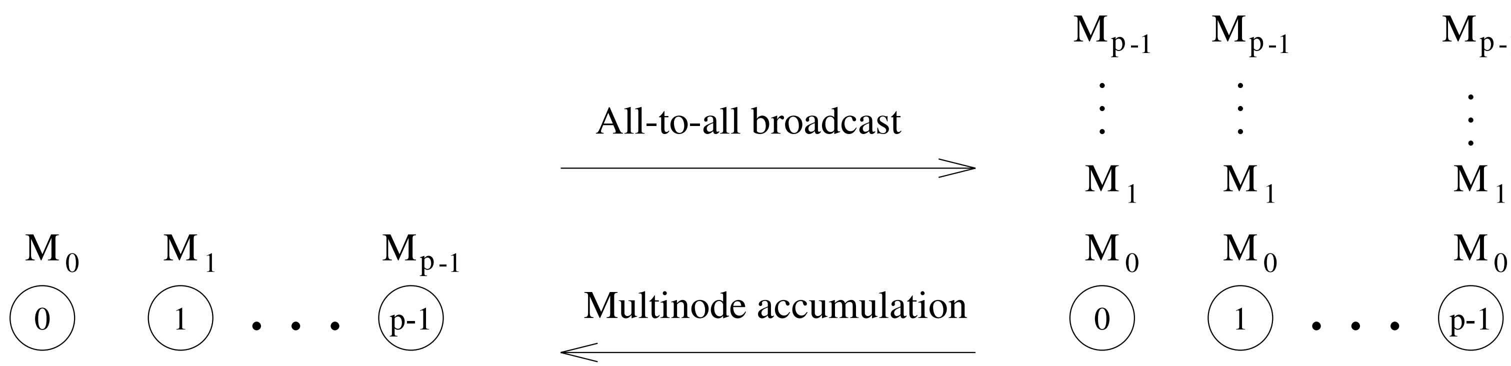
- At the start of a multinode broadcast each processor has m words of data; at the end each processor has a copy of the m words that originated at each of the other processors

- The dual of this operation is a all-to-all reduction or a multinode reduction

- At the start of a multinode reduction each processor has m words of data, the reduction combines all the data from processors using an associative operator to produce m words that are available at all the processors

- Naive multinode broadcast or reduction using p singlenode broadcasts

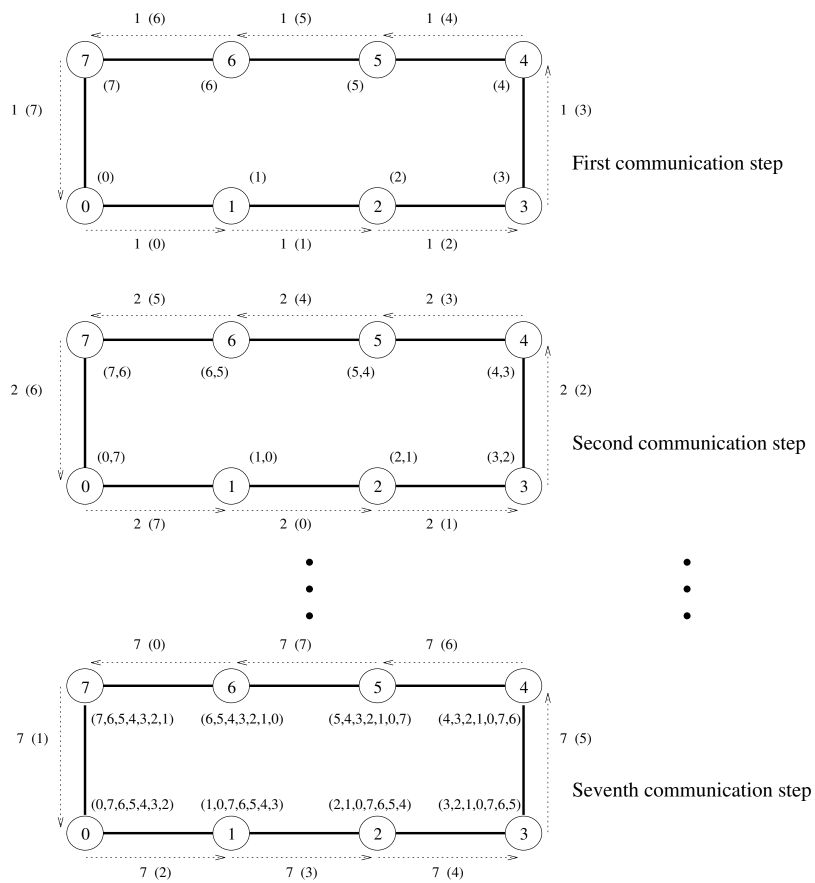
All-to-all Broadcast



Store-and-forward Routing on Ring

- Every processor sends its message to the next processor on the ring in the first step
- In every subsequent step, all processors receive a message from the previous processor and send it to the next processor after retaining a copy for themselves
- $p - 1$ steps
- $(t_s + t_w m)(p - 1)$ cost

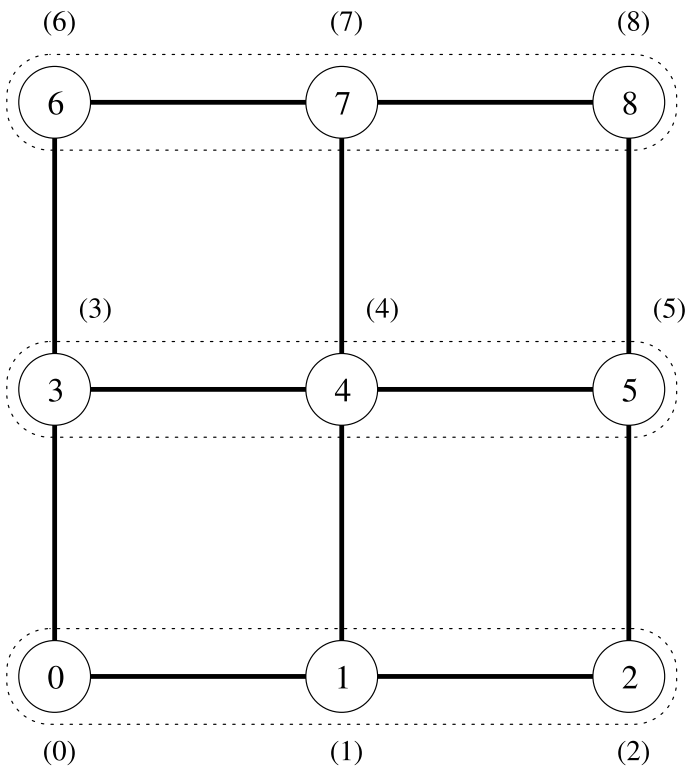
Store-and-forward Routing on Ring



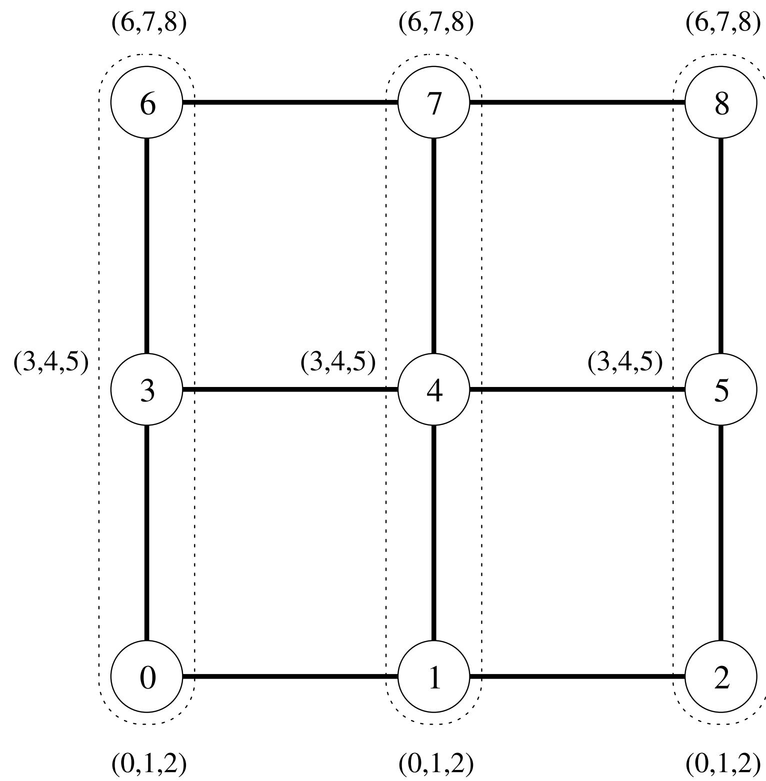
Store-and-forward Routing on 2d Torus

- Use ring method for each row of the torus; compose all $\sqrt{p}$ messages received into a single message and use the ring method for every column
- $2(\sqrt{p}-1)$ steps
- $2t_s(\sqrt{p}-1)+t_w m(p-1)$ cost

Store-and-forward Routing on 2d Torus



(a) Initial data distribution

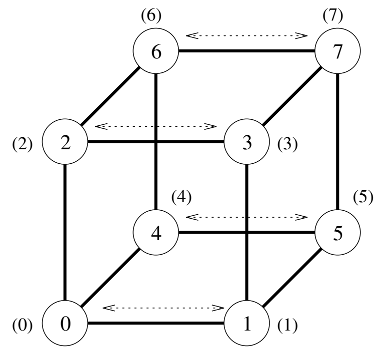


(b) Data distribution after rowwise broadcast

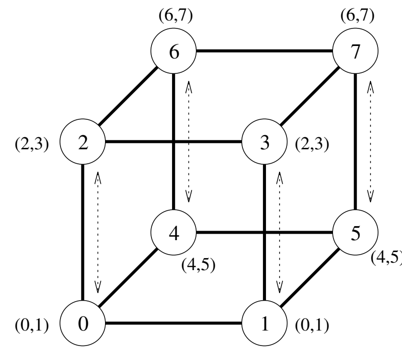
Store-and-forward Routing on Hypercube

- Takes $\log(p)$ steps for a p processor hypercube
- In the ith step, every processor exchanges messages with the neighbouring processor that differs in the ith most significant bit. At each step larger messages are built out of smaller messages for subsequent steps
- $t_s \log(p) + t_w m(p-1)$ cost

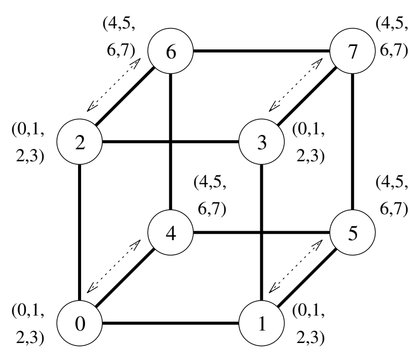
Store-and-forward Routing on Hypercube



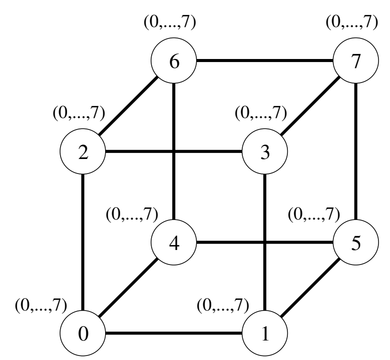
(a) Initial distribution of messages



(b) Distribution before the second step



(c) Distribution before the third step



(d) Final distribution of messages

Cut-through Routing

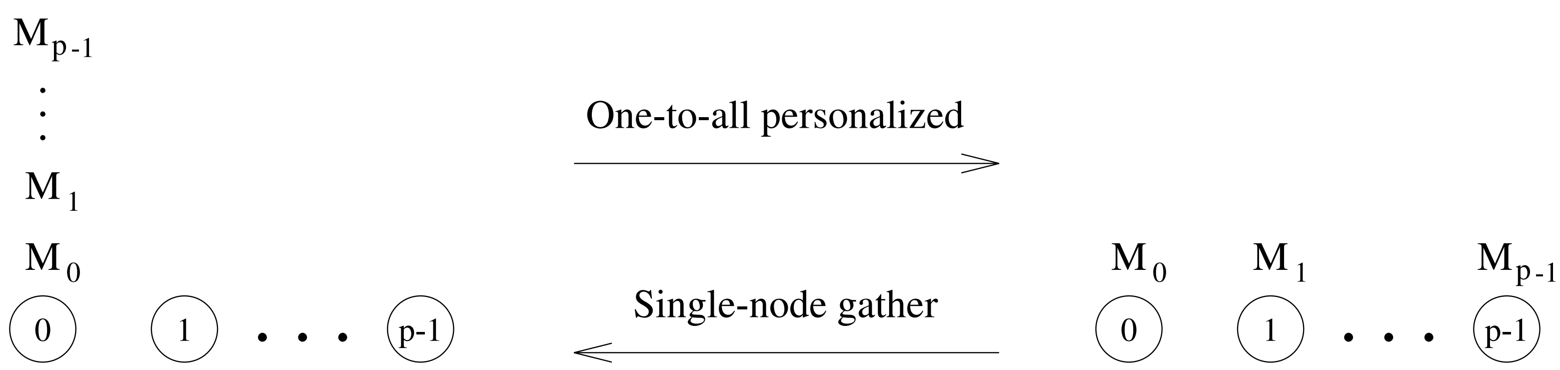
- Cut-through routing does not provide any benefits over store-and-forward for all-to-all broadcasts

- The one-to-all algorithm used for the ring and torus creates contention in the all-to-all case which renders it useless

One-to-all Scatter

- Algorithms often require a processor to send different data to each of the other processors or each of a subset of processors
- This operation is called a one-to-all personalized communication or singlenode scatter
- At the start of a singlenode scatter, the source processor has $p-1$ messages of m words each that need to be sent to each of the other processors; at the end the other processors each have m words
- The dual of this operation is a all-to-one personalized communication or singlenode gather
- At the start of a singlenode gather each processor has m bytes of data, the gather combines all the data from processors to produce $m(p-1)$ words at the receiver
- Naive singlenode scatter or gather using $p-1$ steps

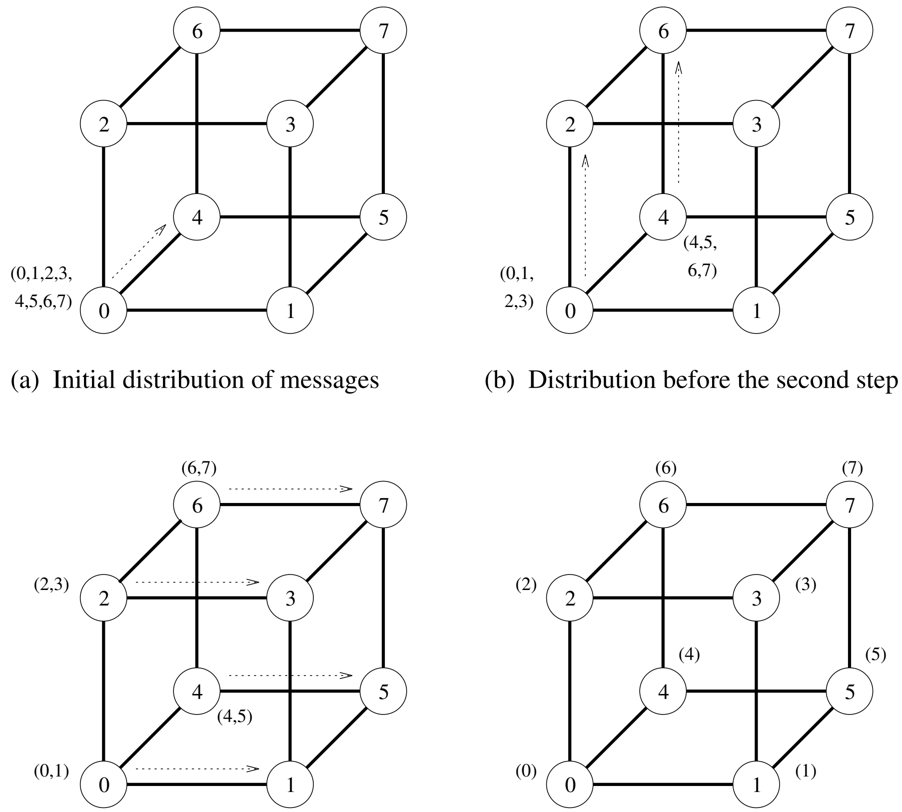
One-to-all Scatter



Store-and-forward Routing on Hypercube

- Takes $\log(p)$ steps for a p processor hypercube
- In the ith step, all processors that have messages transmit half of them to the neighbouring processor that differs in the ith most significant bit
- $t_s \log(p) + t_w m(p-1)$ cost

Store-and-forward Routing on Hypercube



(a) Initial distribution of messages

(b) Distribution before the second step

(c) Distribution before the third step

(d) Final distribution of messages

All-to-all Scatter

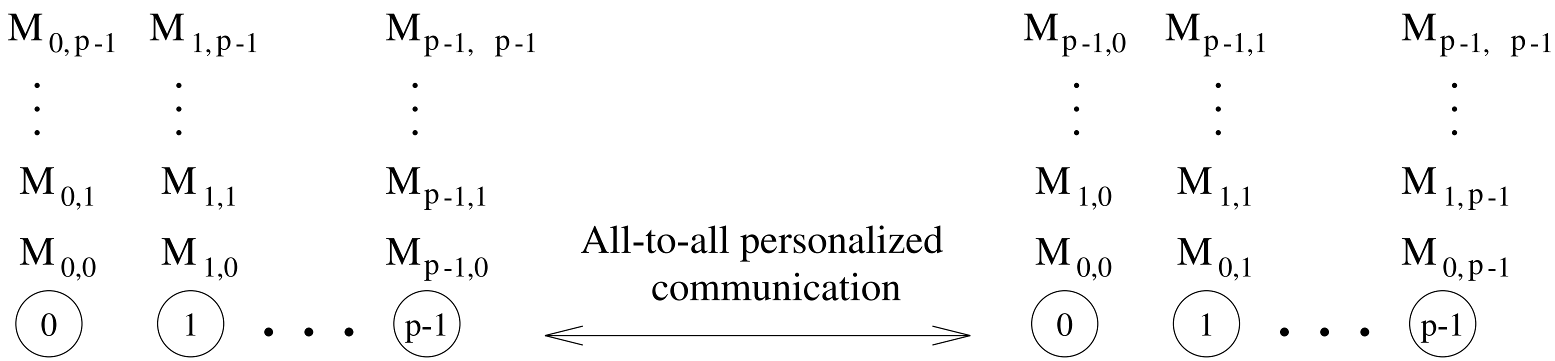
- Algorithms often require each processor to send different data to each of the other processors or each of a subset of processors

- This operation is called a all-to-all personalized communication or a multinode scatter

- At the start of a multinode scatter each processor has $(p-1)m$ words of data; at the end each processor has a copy of the m words that originated at each of the other processors, $(p-1)m$ words in all

- Naive multinode scatter using p singlenode scatters

All-to-all Scatter



Store-and-forward Routing on Ring

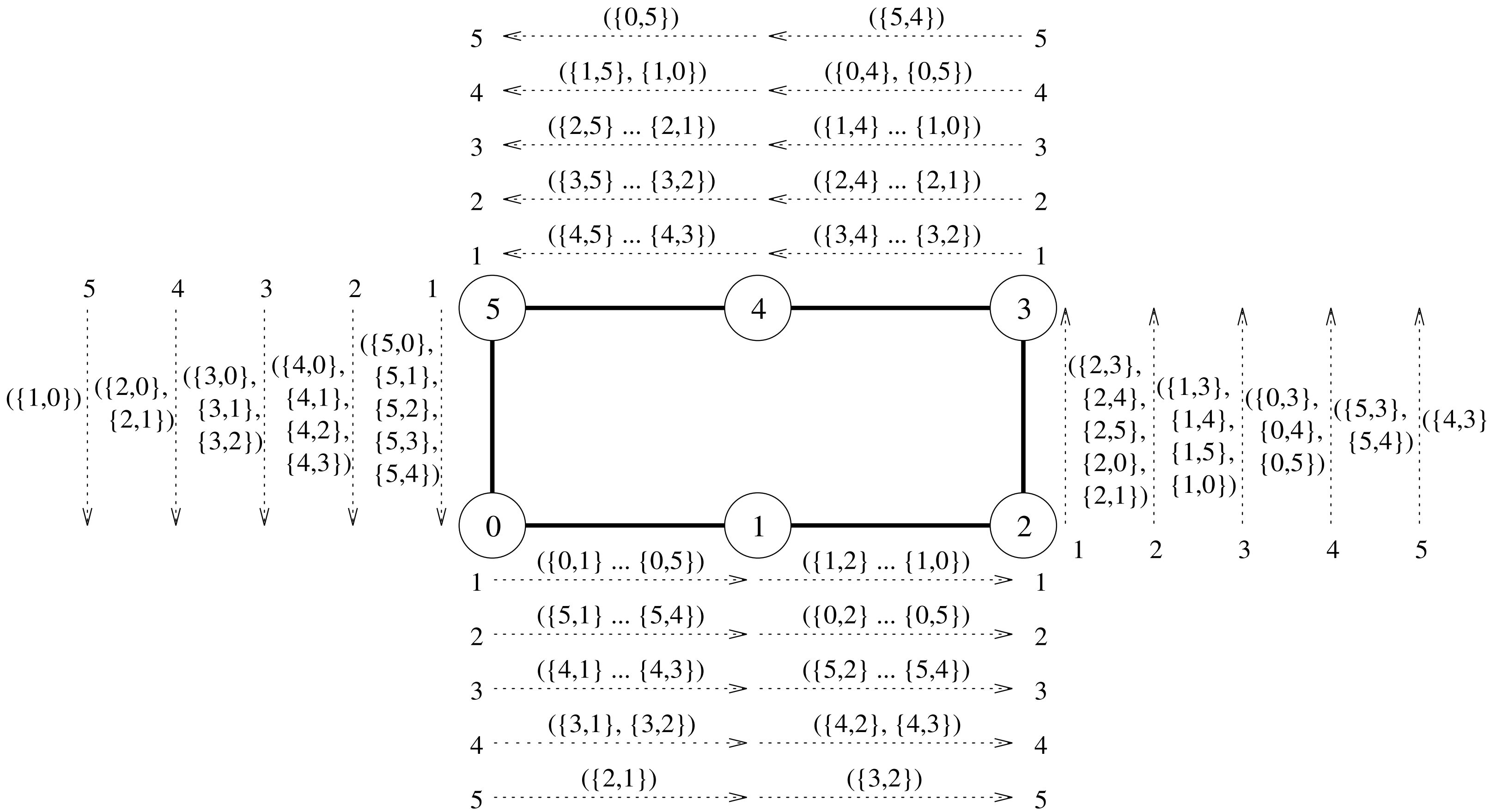
- Every processor consolidates all the data to be sent and sends a single message to the next processor on the ring in the first step

- In every subsequent step, processors retain part of the message received for themselves and send the rest to the next processor on the ring

- $p-1$ steps

- $(t_s + \frac{1}{2}t_w mp)(p-1)$ cost

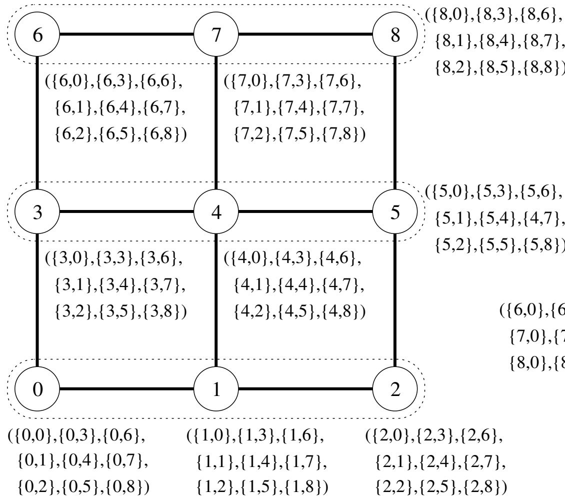
Store-and-forward Routing on Ring



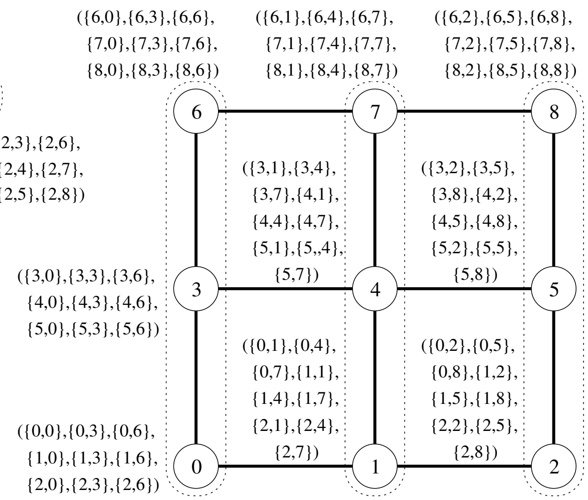
Store-and-forward Routing on 2d Torus

- Processors consolidate messages into groups meant for each processor column
- The ring method is applied to each processor row independently
- Now the messages in each processor are sorted into groups meant for each processor row
- The ring method is applied to each processor column independently
- $2(\sqrt{p}-1)$ steps
- $(2t_s + t_w mp)(\sqrt{p}-1)$ cost

Store-and-forward Routing on 2d Torus



(a) Data distribution at the beginning of first phase

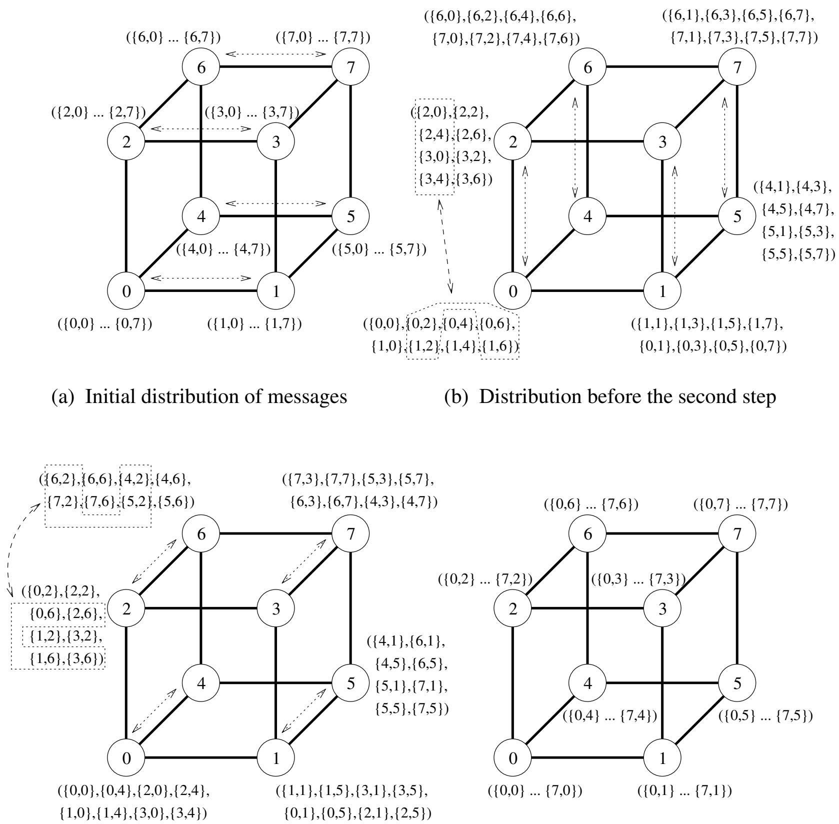


(b) Data distribution at the beginning of second phase

Store-and-forward Routing on Hypercube

- Takes $\log(p)$ steps for a p processor hypercube
- In the ith step, every processor exchanges messages with the neighbouring processor that differs in the ith least significant bit
- At each stage, every processor holds p messages, $\frac{p}{2}$ of these are consolidated into a single message for exchange with a neighbouring processor
- $(t_s + \frac{1}{2} t_w m p) \log(p)$ cost

Store-and-forward Routing on Hypercube



(a) Initial distribution of messages

(b) Distribution before the second step

(c) Distribution before the third step

(d) Final distribution of messages

Summary

- One-to-all Broadcast
- All-to-all Broadcast
- One-to-all Scatter
- All-to-all Scatter