

---

## Parallel Algorithms

---

*Due date: January 21st, 2014 before class!*

### Problem 1 (10 Points)

Consider greedy routing on the hypercube with random intermediate locations, i.e. if a packet at node  $i$  has to be sent to node  $j$ , the routing protocol first sends it to a random node  $r$ , then sends it from  $r$  to  $j$ . In this case, routing paths can be as long as  $2d$ , where  $d$  is the dimension of the hypercube. Now, consider the following variation: for every packet and its random intermediate location  $r$ , we compare the lengths of the greedy paths  $i \rightarrow \dots \rightarrow r \rightarrow \dots \rightarrow j$  and  $i \rightarrow \dots \rightarrow \bar{r} \rightarrow \dots \rightarrow j$  (with  $\bar{r}$  having the encoding of  $r$  with all bits flipped), and route the packet along the shorter path.

Show that the routing paths are shorter in the worst case.

### Problem 2 (20 Points)

A *packing* problem consists of routing any collection of  $m \leq n$  packets contained in level  $n$  of a log  $n$ -dimensional butterfly to the first  $m$  nodes in level 0 of the butterfly such that the relative order of the packets remains unchanged.

1. Consider removing all nodes (and incident edges) from the leftmost or rightmost level of a butterfly, respectively. Which structure have the remaining networks?
2. It is possible that a processor that contains a packet in a packing problem may not know the correct destination for the packet. How can you figure out the correct destinations for the packets with only two runs through the butterfly?  
*Hint:* Use a parallel prefix.

3. Show that after the first step of the greedy packing protocol, there are no collisions, i.e. there are no two packets sent to the same node.

*Hint:* Consider two neighboring packets. What is the difference in their corresponding destinations?

Keep in mind that for this case, we have a reversed butterfly at hand, as seen in Fig. 1. Would this work for a normal butterfly and why (not)?

4. Complete the proof that there are no collisions on any level by induction.

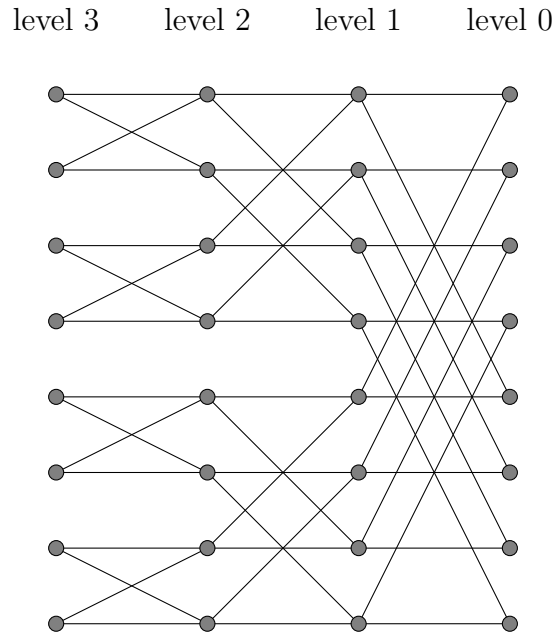


Figure 1: reverse butterfly

### Problem 3 (10 Points)

A *spreading* problem consists of routing a contiguous set of  $m \leq n$  packets from level 0 of a  $\log n$ -dimensional butterfly to any collection of  $m$  destinations at level  $\log n$  such that the relative order of the packets remains unchanged.

A *monotone* routing problem is one where the relative order of the packets remains unchanged.

1. Show how to greedily route any spreading permutation with a congestion of 1.
2. Show how to greedily route any monotone permutation with a congestion of 1.