Dr. Christian Scheideler
Johns Hopkins University

Baltimore, Oct. 29 2003

Theory of Network Communication

Fall 2003

Assignment 7

Problem 15 (4 points)**:**
Show that there is a caching strategy for the hypercube that is $O(\log n)$ competitive.

Hints: Use a proof similar to the one for the mesh. Recall the oblivious routing strategy we used for the hypercube and the fact that a d-dimensional hypercube contains two $(d-1)$-dimensional hypercubes.

Problem 16 (6 points)**:**
Write a C++ program in the Spheres environment that caches data in an $n \times n$-mesh as shown in the lecture but *only* allows read requests (to simplify your task). Generate a sequence of 5 read requests for a single data item from different nodes in the 10×10-mesh to test your program. Print out the program and the result of the test.

Hints: Again, forget about sending sync packets around. Just do method invocations.

Remember that initially only the root of the decomposition tree stores the copy. All sign posts initially show upwards (this may have a default value of 0). To find the closest tree node having a copy, we follow the sign posts. You can identify a node in the decomposition tree uniquely by its level and the node in the lower left corner of the corresponding submesh, and you can use this to compute a hash value (using the SHA-1 hash function available from the web page of the course) to determine which node in the submesh serves as the representative of the tree node.

You have to implement a couple of methods. Read to find the nearest copy of the data item in the tree, Reply to send a reply back in the tree (remember placing copies!), and XYRoute to simulate sending a read or reply request along a tree edge by routing a message carrying the read or reply request through the mesh according to the $x - y$ routing strategy.