

Interconnection Networks Summative

wcrr51

2021

Part 1

Question 1

1. Pre-building routes and storing them in routing tables at the nodes.

One of the primary overheads when storing the routing tables at nodes is that this uses storage space, meaning that, for any arbitrary (or naive) interconnection network, each node must store the route for (or at minimum the next node in the route) for any other node, meaning the table size is between linear and quadratic in the number of nodes. This leads to the primary difficulty when the interconnection network must be fault-tolerant in that if a node goes offline, all the other nodes (and their respective routing tables) will be ignorant of this change, meaning some routes may be blocked (and thus packets are lost). In this case, to remain fault-tolerant, one must use additional resource overhead to detect faults and correspondingly update the routing tables for all nodes (preventing program-specific messages being sent).

2. Building routes on-the-fly.

If the neighbours of a faulty node are aware of the issue, they can dynamically adjust their route to avoid the node, and thus avoid lost packets. However, building routes on-the-fly carries additional processing overhead, as a message arrives at or is generated by each node, some processing power (and time) needs to be spent working out which node it should be sent to next. While this processing is taking place, the message is stationary meaning higher latency. The main difficulty with building routes on-the-fly is that the interconnection network needs to be well-described (as opposed to arbitrary/random), which incurs extra design complexity (especially when incorporating path diversity for fault-tolerance).

Question 2

(a)

Using the fact that the network is symmetric, the table will be the same size for all the nodes.

1. There are 4 nodes of minimal distance 1 from the source, requiring $4 \cdot 4 \cdot 1 = 16$ bits.
2. There are 6 nodes of minimal distance 2 from the source, requiring $6 \cdot 4 \cdot 2 = 48$ bits.
3. There are 4 nodes of minimal distance 3 from the source, requiring $4 \cdot 4 \cdot 3 = 48$ bits.
4. There is 1 node of minimal distance 4 from the source, requiring $1 \cdot 4 \cdot 4 = 16$ bits.

Summing this up, there are 15 nodes (excluding the source) requiring 128 bits to be stored.

Interestingly, as an aside, the recursive nature of the n -cube means that the number of nodes of minimal distance i from a node is $\binom{n}{i}$. Hence, the above expression can be calculated with $4 \sum_{i=0}^n i \binom{n}{i}$ which, for $n = 4$, $4 \sum_{i=0}^4 i \binom{4}{i} = 4 \cdot (0 + 4 + 16 + 16 + 4) = 4 \cdot 32 = 128$ bits.

(b)

In a Q_4 there exist $2^4 = 16$ nodes, thus 16 table entries. As each table entry represents one node, 4 bits are required to store the target, and 4 bits are required to store the next node.

Hence, for a table consisting of the next path required to get to all nodes, $16 \cdot 8 = 128$ bits. As each node need not store a table entry for itself, and assuming neighbouring nodes may be routed to through other neighbours (i.e. the fastest path to a neighbour is not necessarily a direct hop to the neighbour), $128 - 8 = 120$ bits are required.

(c)

Question 3

(a)

(b)

(c)

(d)

Part 2

Part 3

Part 4

Part 5