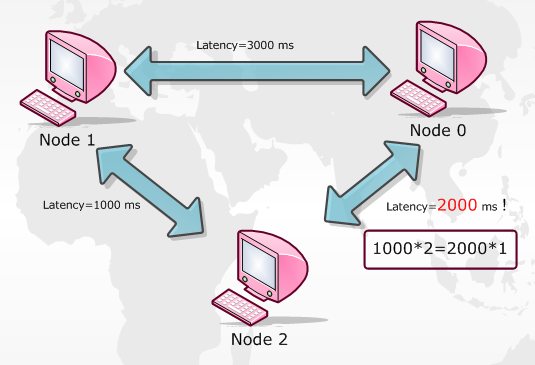
We find a way to ensure that in a linearization the write of value i is the last of the 3 write operations. First, assume every node executes exactly the same sequence of operations, then the node which is slowest (enough slow) among others will always succeed in writing the last value, by slowest we mean the node have the largest delay (enough delay) on links to others.



Figure

For instance, consider the topology shown in Figure 1, each node pi executes D500:Wi:R:D500:R, in writing phase of node 2 (node 2 writes 2), it first broadcast to ask for the largest timestamp in the network, sooner it receive 0 from itself and 0 from node 1, since the number reaches the majority, node 2 can write 2 to local value and increases its local timestamp to 1, so does node 1. Meanwhile, node 0 wants to write 0, it first broadcast to ask for the largest timestamp as well, if this message reaches node 2 after node 2 increases its timestamp, then node 2 will reply timestamp value 1 to node 0, therefore node 0 is able to send its writing value 0 to all others using the largest timestamp so far, and other nodes have to adopt it (overwrite its local value if any). In conclusion, the latency between node 0 and node 2 should be at least greater than the latency of round trip between node 1 and node 2, or > 1000 \* 2 = 2000 so to speak. Based on such configuration, we can put whatever node to the top-right position in Figure 1 to let it write the last value.