

4 Concurrent and Distributed Systems (mk428)

- (a) Describe briefly what is meant by *consensus* in distributed systems, and how it relates to *total order broadcast*. [3 marks]
- (b) Define the five properties that a total order broadcast algorithm must specify. For each property, state whether it is a safety or a liveness property. [5 marks]
- (c) Your colleague proposes implementing a total order broadcast system as follows. A server keeps an in-memory array of references to messages, initialised with nulls. When a client wants to broadcast a message, the client sends this message over the network to the server. On receipt, a thread on the server performs an atomic compare-and-swap operation that updates the first null array entry to point at that message, provided that another thread has not concurrently updated the same array entry. All clients then deliver messages from non-null array entries in the order they appear in the array.

Is this a correct total order broadcast algorithm according to your definition from Part (b)? Justify your answer. [4 marks]

- (d) How does the algorithm from Part (c) compare to Raft? [3 marks]
- (e) Imagine you have a fault-tolerant counting service, which can be queried by clients. Every query atomically increments a counter and returns the new counter value. Can this service be used to implement a total order broadcast system according to your definition from Part (b)? If yes, explain how this is done. If no, explain why it cannot be done. [3 marks]
- (f) In the scenario of Part (e), does it make a difference if you assume that there are no more than two clients? Justify your answer. [2 marks]