5 Concurrent and Distributed Systems (mk428)

You are designing *Warbler*, a social networking service in which users can post public messages called *chirps*. A chirp may or may not be a reply to another chirp. Moreover, one user can *follow* another user, which means subscribing to the messages they post.

(*a*) Write pseudocode for three operations: (1) user u_1 following user u_2; (2) u_1 unfollowing (ceasing to follow) u_2; and (3) obtaining the current set of followers of some user u. These operations must be fault-tolerant: you should store the data on five servers, and the operations must be able to complete successfully as long as any three or more servers are available. Your algorithm should ensure read-after-write consistency.

Assume an asynchronous system model with fair-loss network links and crash-recovery faults. Write your algorithm in terms of messages sent over point-to-point (unicast) links. You may assume that the user is already authenticated (e.g. their password has been checked), and you may assume that users' client software can generate Lamport timestamps.

Start by describing the nodes in the system, the structure of the data stored at each node, and the form and purpose of messages exchanged by nodes. Then give the pseudocode for the operations listed above. [12 marks]

(*b*) When a user posts a chirp, it needs to be sent to all of that user's followers. Write pseudocode to do this, using the operation for getting a user's followers from part (*a*). Chirps must be delivered to a given recipient in the following order: chirps posted by the same user should be delivered in the order they were posted; and when one chirp is a reply to another, the reply should be delivered after the chirp it is replying to. Apart from these rules, the chirps may be delivered in any order. [8 marks]

In both parts, in addition to the pseudocode, please also briefly explain and justify the design of your algorithm, and specify any assumptions you are making.