The University of Melbourne
School of Computing and Information Systems
COMP90020 Distributed Algorithms

Tutorial Week 7: Multicast

Notes

Solutions

30. In the case of open groups a process may leave and join freely as opposed to closed groups which define static membership.

Agreement: unchanged
Integrity: unchanged
Validity: if a process multicasts(m), then some member of group(m) will eventually deliver m.

31.

(a) Reversing the order of those lines means that a process can deliver a message and then crash before sending it to the other group members – which might, in that case, not receive the message at all. This contradicts the uniform agreement property.

(b) The reliable multicast algorithm based on IP multicast does not satisfy uniform agreement. A recipient delivers a message as soon as it receives it; if the sender was to fail during transmission and that same message was note to have reached the other group members then the uniform agreement property would not be met.

32. Consider two groups g1, g2. Where g1 intersection g2 = {p2,p1} p1 first FO-multicasts m1 then FO-multicasts m2. In this case p2 will need to order the delivery of m1,m2 where if m2 is delivered, then m1 must be delivered before m2. This is not defined for multiple groups. We can simply sequence messages for each group on send. e.g. keep a count of the number of sends to that group and pass on all counts with each message.

33. In a group, there are eight members, of which at most one can crash at any time. You have no idea about which process can crash. The topology is a completely connected network.

(a) Number processes 0,...,7. 0 Initiates the multicast. No way to detect crash failures reliably (async) So assuming the worst case no process crashes. So 7 Initial messages (not counting self-msg) and for each receiving process they must broadcast to the other 6 (we don't inlude the initial sender) 7 * 6 = 49

(b) Consider an extension of the previous, a node cannot know if its the only member that delivered the message just before the sender failed. Once any node receives a message, it only needs to multicast it once to every other node (excluding sender) it is the same as above. 7 + 7 * 6 = 49

34.

(a) Suppose p TO-multicasts m1 which q receives, then q TO-multicasts m2. The sequencer must order m2 after m1. So all correct processes must also deliver m1 before m2. Suppose p TO-multicasts m1, then TO-multicasts m2. As B-multicast is FIFO, m1 must be sequenced before m2. So all correct processes must also deliver m1 before m2.

(b) Generally yes, so long as the implementation of total ordering guarantees that sequence numbers of any message sent is greater than that of any received by the sending process. In which case FIFO-ordering enforces this property.

35. Consider a multiparty game of quiz involving five teams.

A form of multicast where answers after 30 seconds are not delivered by any process.Time-multicast. with delta = 30.

36.

(a) Reliable version of TO-multicast (using R-multicast instead of B-multicast)

(b) Using (a) we can continue counting in the event of a partition and by the guarantees of RELIABLE and TOTAL ORDER it must be the case that the underlying protocol will ensure delivery if one partition does.