2PC and DHT exercises

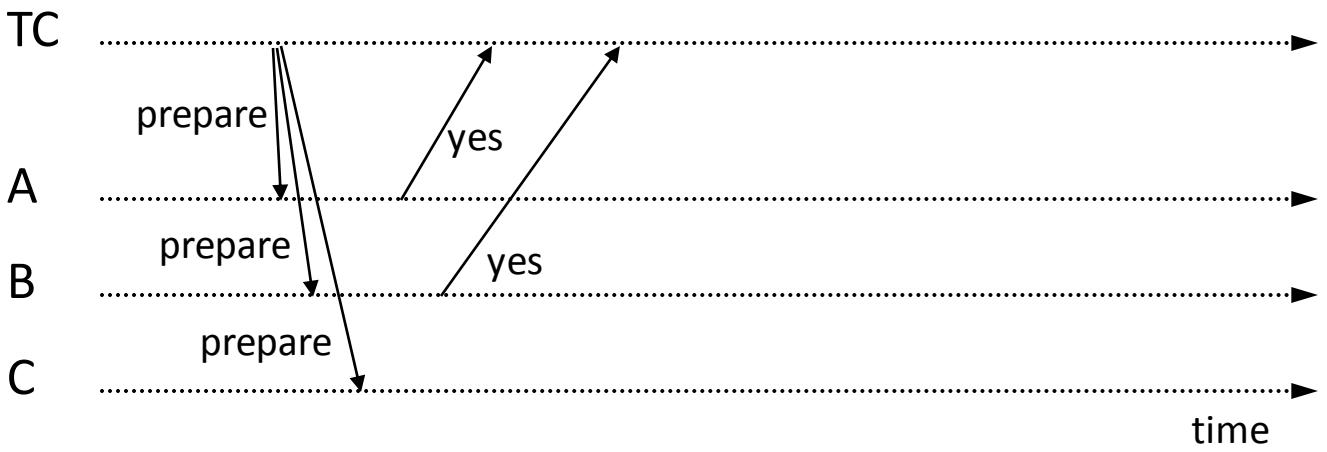
2PC exercises

Q1: The state of a 2PC protocol is given below. TC has not yet received a "yes/no" (prepare-ready/prepare-abort) message from node C.

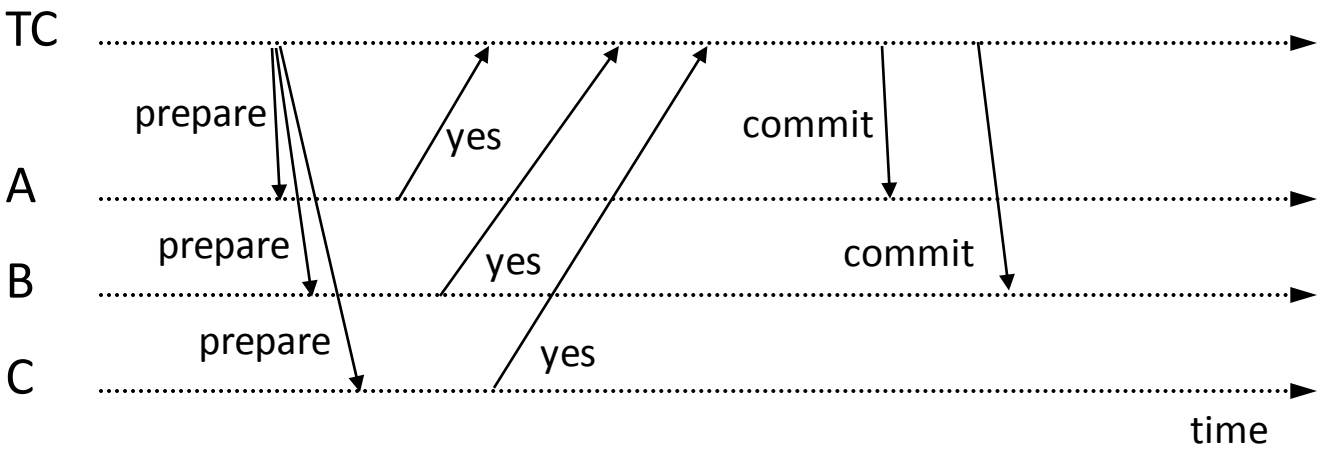
a) Would it be correct for the TC to abort the transaction at this stage?

b) Would it be correct for the TC to commit the transaction at this point?

Explain why or why not, and what would the follow-up messages to complete the 2PC protocol be.



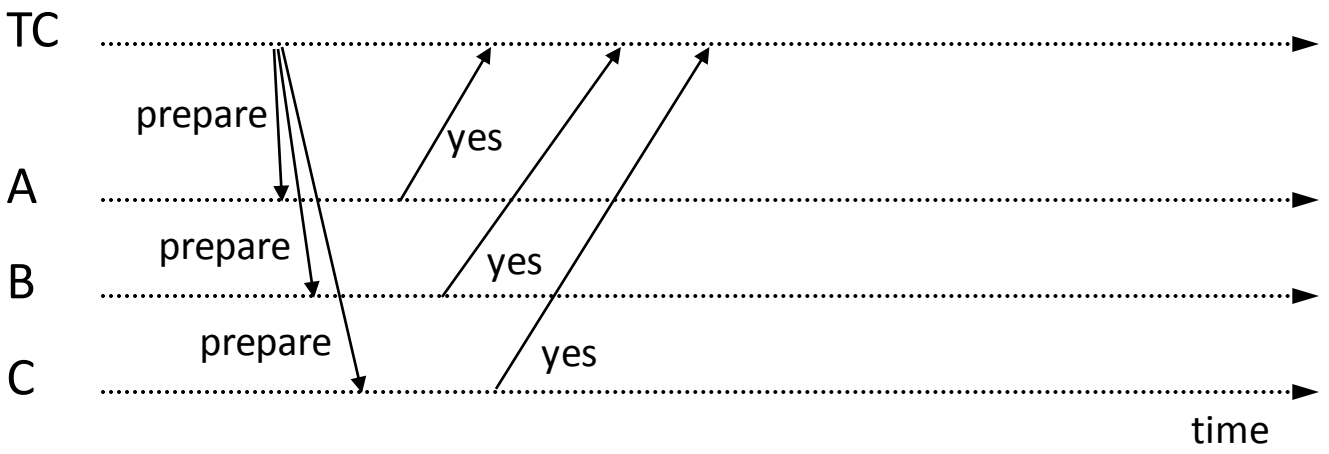
Q2: The state of a 2PC protocol is given below. Would it be correct for node C to execute its part of the transaction at this stage, as if it had received a commit message (similar to A and B)? Explain why or why not. (In phase 2, assume C contacts node B and discovers that node B has received a commit message.)



Q3: The state of a 2PC protocol is given below. In phase 2 assume node C contacts node B and discovers that node B has not yet received a commit or abort. Node C cannot reach node A.

a) Would it be correct for node C to commit or abort at this stage? Explain why.

b) Describe a sequence of events that could lead A to commit, but not B and C.



DHT exercises

(extracts from the book)

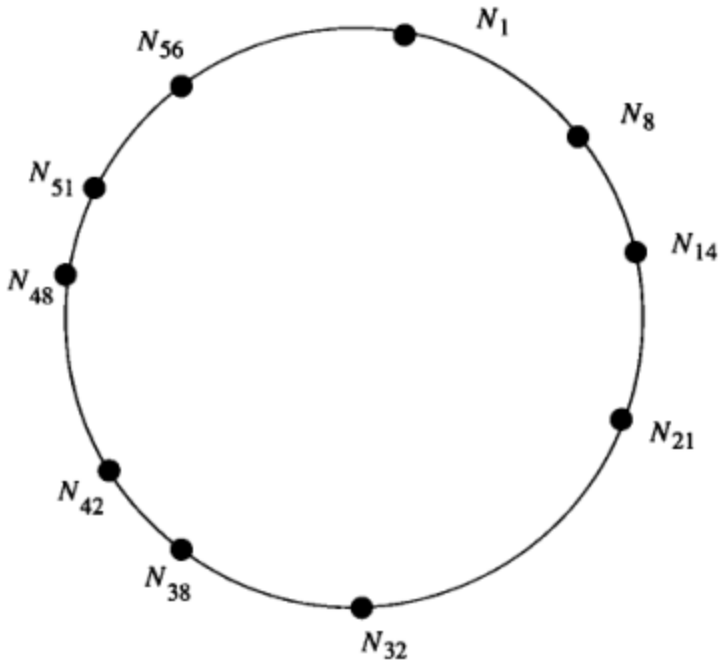


Figure 20.14: A chord circle

To place a node in the circle, we hash its ID i, and place it at position h(i). We shall henceforth refer to this node as $N_{h(i)}$. Thus, for example, in Fig. 20.14, N_{21} is a node whose ID i has h(i) = 21. The successor of each node is the next higher one clockwise around the circle. For example, the successor of N_{21} is N_{32}, and N_1 is the successor of N_{56}. Likewise, N_{21} is the predecessor of N_{32}, and N_{56} is the predecessor of N_1. The nodes are located around the circle using a hash function h that is capable of mapping both keys and node ID's (e.g., IP-addresses) to m-bit numbers, for some m. In Fig. 20.14, we suppose that m = 6, so there are 64 different possible locations for nodes around the circle. In a real application, m would be much larger. Key-value pairs are also distributed around the circle using the hash function h. If (K; V) is a key-value pair, then we compute h(K) and place (K; V) at the lowest numbered node N_j such that h(K) ≤ j. As a special case, if h(K) is above the highest-numbered node, then it is assigned to the lowest-numbered node. That is, key K goes to the first node at or clockwise of the position h(K) in the circle.

The circle of nodes of Fig. 20.14 (previous slide) is given.

Q4: Where do key-value pairs reside if the key hashes to the following numbers?
(a) 35
(b) 20
(c) 56
(d) 60

Q5: Construct the finger tables for:
(a) N14
(b) N51

Q6: What is the sequence of messages sent if:
a) N14 searches for a key that hashes to 27?
b) N8 searches for a key that hashes to 5?
c) N51 searches for a key that hashes to 43?