Computer Systems
Quiz 7

Question 1

What are the smart contracts, bundling code and data, on the Internet Computer called?

a) Containers.

b) Carriers.

c) Canisters.

d) Capsules.

Question 2

"A _is a set of nodes providing state replication for the canisters deployed on it."

a) BLS set.

b) Subnet.

c) Subset.

d) Xnet.

Question 3

How many subnets execute a specific canister, if the total number of nodes of the Internet Computer is n?

a) 13

b) n

c) 40

d) All previous answers are wrong.

Question 4

Which of the following properties does abortable broadcast guarantee?

a) Totality.

b) Agreement.

c) Validity.

d) Weak integrity.

Question 5

How can totality be achieved for a message that is never aborted when using abortable broadcast?

a) By calling broadcast on the message upon delivery.

b) By verifying that the sender is correct.

c) By adding a delay before delivering the message.

d) All previous answers are wrong.

Question 6

In the abortable broadcast algorithm with a slot table capacity of C, how many periodic retransmission messages does a newly joined node in a subnet with n nodes need to receive to catch up?

a) $O(C)$

b) $O(n)$

c) $O(n^2)$

d) $O(nC)$

Question 7

Which of the following statements about the random beacon mechanism is correct?

a) Any $f + 1$ random beacon shares yield the same next random beacon.

b) Sets of $f + 1$ random beacon shares may yield different random beacons but that's okay because each set contains at least one correct node.

c) Sets of $2f + 1$ random beacon shares are used to ensure that the sets intersect in at least one correct node.

d) $f + 1$ random beacon shares are used instead of $2f + 1$ to reduce the communication complexity.

Question 8

Which statement about the notarization and finalization mechanism is correct for blocks of the same round?

a) If a block is notarized, no other block can be finalized.

b) If a block is notarized, no block from a block maker with lower priority can be notarized.

c) If a block is finalized (through $2f + 1$ finalization shares), no other block can be notarized.

d) If a block is finalized, there must have been a period of synchronicity.

Question 9

What is the benefit of checking ingress message expiry before notarizing a block?

a) Later expiry checks can be omitted.

b) No block space is wasted for expired messages which won't be executed anyway.

c) The block maker doesn't have to conduct the check.

d) All answers are correct.

Question 10

Why is the certified height included in block validation?

   a) To detect skipped messages from other subnets.

   b) To identify non-determinism.

   c) To ensure a feedback loop between consensus and message routing.

   d) All answers are correct.

Question 11

How do the nodes determine that a received cross-subnet (xnet) message is valid?

   a) The nodes query the network nervous system.

   b) The nodes query the nodes on the source subnet for validation data.

   c) The nodes verify the subnet signature on the xnet message.

   d) The nodes trust all messages from other subnets.

Question 12

What is the advantage of only executing messages in finalized blocks?

   a) Forks can be detected reliably and handled in canister code.

   b) Executed messages never need to be rolled back.

   c) Since the finalization comprises $2f + 1$ signature shares, the nodes only need to wait for f more responses to fully confirm the block.

   d) There is no advantage to finalizing before executing.

Question 13

How does a node know which protocol version to run?

   a) By asking other nodes in the subnet.

   b) By querying a canister of the subnet it belongs to.

   c) By querying an NNS canister.

   d) Any protocol version works.

Question 14

How can a user check if the Internet Computer processed a message correctly?

   a) By verifying the signature on the response.

   b) By gathering all ingress message payloads since genesis and replaying them.

   c) By reading the canister code.

   d) All previous answers are wrong.

Question 15

Given a subnet consisting of 25 nodes, how many faulty (crashed, malicous, buggy, ...) nodes can the Internet Computer protocol tolerate?