Exam Distributed Algorithms

Vrije Universiteit Amsterdam, 31 May 2023, 18:45-21:30

(You may use the textbook Distributed Algorithms: An Intuitive Approach*. Use of slides, solutions to exercises, notes, laptop, calculator is* not *allowed.)*

(The exercises in this exam sum up to 90 points; each student gets 10 points bonus.)

1. Let the Dijkstra-Scholten algorithm be employed to detect termination of some centralized basic algorithm. Give an execution of the basic algorithm in which an active process q has a parent p in the Dijkstra-Scholten tree while q was made active for the last time by a process $r \neq p$. (10 pts)

   Solution: Let the network consist of channels pq and qr. First, q is made active by a basic message from initiator p, so that q joins the tree with parent p. Next, q makes r active by sending it a basic message, so that r is q's child in the tree. Next, q becomes passive, but remains in the tree because it has child r. Next, q is made active again by a basic message from r.

2. Consider the weight-throwing termination detection algorithm with the counter $credit_p$ for recording weight, to avoid underflow. Why does an active process that receives a basic message not return this weight to the initiator immediately, but only after it has become passive? (10 pts)

   Solution: Since weight cannot be reused, there is no hurry in returning weight to the initiator, who will need to keep returned weight separate from weight it distributes through the system. Returning weight to the initiator after having become passive, instead of immediately, reduces overhead in the number of control messages.

3. Consider Franklin's election algorithm for undirected rings (with non-FIFO channels).

   (a) Give an example to show that an active process in election round n can receive a message for round $n + 1$ before receiving the message for round n

from this same direction, where these two messages carry different IDs. (8 pts)

Solution: Consider four consecutive active processes in the ring with IDs 1, 4, 2, and 3, respectively, that send out their messages in round 0. Processes 2 and 4 receive the messages for election round 0, where 2 becomes passive while 4 moves to round 1 and sends out messages again. The message from process 4 in round 1 is forwarded by process 2 and overtakes the message from process 2 in round 0. Now process 3 receives the message from process 4 in round 1 before the message from process 2 in round 0.

(b) Argue that it cannot receive a message for two rounds ahead. (12 pts)

Solution: Let active process p be in election round n. Consider in either of the two directions the nearest neighbor q of p that reached an election round beyond n. (If there is no such neighbor, clearly p can never have received a message for a round beyond n.) Then q can from its direction toward p not have received a message for an election round beyond n, so it must be in round $n+1$. Hence it cannot have sent a message toward p in a round beyond $n+1$.

4. Consider the Bracha-Toueg k-crash consensus algorithm, with $k < \frac{N}{2}$. Let more than $\frac{N+k}{2}$ processes choose the value b in the initial configuration. Argue that the correct processes will inevitably decide for b within three rounds. (10 pts)

Solution: In round 0, each correct process receives more than $\frac{N-k}{2}$ b-votes, and only $(1-b)$-votes with weight 1. So it changes its value to b.

In round 1, each correct process receives only b-votes, and changes its weight to $N-k$.

In round 2, each correct process receives only b-votes with weight $N-k$. So each correct process decides for b no later than round 2.

5. Consider the heights (h_1, h_2) in the Walter-Welch-Vaidya mutual exclusion algorithm.

(a) Argue that the minimum h_1-value in the network never decreases during computations. (8 pts)

Solution: There are two situations in which a nonroot p updates the h_1-value of its height.

In the first case, all p's edges have become incoming. This means p's original h_1-value is greater or equal than the h_1-values of all its neighbors. And p increases its h_1-value to the minimum h_1-value among its neighbors plus 1.

In the second case, p becomes the new root. This means p's original h_1-value is greater or equal than the h_1-value of the old root. And it copies the h_1-value of the old root.

In both cases, clearly, p's new h_1-value is greater or equal than the minimum h_1-value among p and its neighbors before p's h_1-value was updated.

(b) Give an example where the minimum h_2-value in the network increases during a computation. (6 pts)

Solution: Let nonroot p have an h_2-value at least two smaller than any other node in the network.

The easiest example is when p becomes the new root. Then p changes its h_2-value to the h_2-value of the old root minus one.

Another example is that the only outgoing edge of p disappears, and p forges a new height which has the same h_1-value as a neighbors of p. Then p changes its h_2-value to the minimum h_2-value among its neighbors with the same h_1-value, minus one.

In both cases, clearly the minimum h_2-value in the network increases.

6. Suppose that in step 2 of the Kerberos authentication protocol, the authentication server would include the server ID S in the ticket it sends to the client. Explain how this would seriously hamper the applicability of the Kerberos protocol. (10 pts)

Solution: A key advantage of the Kerberos protocol is that, by separating the authentication and ticket-granting server, a user can reuse a session key and ticket from the authentication server at the ticket-granting server for setting up sessions with different servers, without having to repeatedly employ her password. If the server name S were included in the ticket obtained from the authentication server, then this reuse would become impossible, as each session with another server would require obtaining a new ticket from the authentication server.

7. (a) Consider the Winternitz signature scheme with $k = 10$ and $\ell = 3$. Let 0100111010 be the hash of Alice's message to Bob. Explain how Alice signs

her message, taking into account the checksum, and how Bob verifies this signature. (8 pts)

Solution: Two 0's are padded at the left of the hash of the message, to make its length divisible by 3. The binary representations of b_1, b_2, b_3 and b_4 are 000, 100, 111 and 010, respectively, so $b_1 = 0$, $b_2 = 4$, $b_3 = 7$ and $b_4 = 2$.

The checksum is $(7-0)+(7-4)+(7-7)+(7-2) = 15$. Its binary representation is 1111. Two 0's are padded at the left to make its length divisible by 3. This means the binary representations of b_5 and b_6 are 001 and 111, so $b_5 = 1$ and $b_6 = 7$.

Alice generates large random numbers $X_1, X_2, X_3, X_4, X_5, X_6$, which form her private key.

Her public key is $h(h^7(X_1) \,\|\, h^7(X_2) \,\|\, h^7(X_3) \,\|\, h^7(X_4) \,\|\, h^7(X_5) \,\|\, h^7(X_6))$.

Her signature is $X_1 \,\|\, h^4(X_2) \,\|\, h^7(X_3) \,\|\, h^2(X_4) \,\|\, h(X_5) \,\|\, h^7(X_6)$.

Bob uses the hash of the message to compute $b_1, \ldots, b_6$.

With regard to Alice's signature, he applies h^7 to $h(X_1)$, h^3 to $h^4(X_2)$, leaves $h^7(X_3)$ unchanged, h^5 to $h^2(X_4)$ and h^6 to $h(X_5)$ and leaves $h^7(X_6)$ unchanged. Finally, he applies h to the concatenation of these strings, compares the result to the public key, and accepts Alice's signature.

(b) Suppose the Winternitz signature from (a) is placed in the third leaf of a binary Merkle tree of depth 4 and used by Alice in a Merkle signature of a message to Bob. Explain concretely what the signature looks like and how this signature is employed by Bob to verify whether the public key is genuine. (8 pts)

Solution: Alice's signature takes the form $sig_3 \,\|\, Y_3 \,\|\, h(Y_4) \,\|\, H_{31} \,\|\, H_{22} \,\|\, H_{12}$ with sig_3 the signature and Y_3 the public key from (a). Each H-values is the h-value of the concatenation of the strings in the two children of the corresponding node in the Merkle tree.

Bob applies h to Y_3 to determine the value in the third leaf. Then he consecutively computes $H_{32} = h(h(Y_3) \,\|\, h(Y_4))$, $H_{21} = h(H_{31} \,\|\, H_{32})$, $H_{11} = h(H_{21} \,\|\, H_{22})$, and $H_{01} = h(H_{11} \,\|\, H_{12})$. He compares the computed value of H_{01} with the Merkle root.