Resit Distributed Algorithms

Vrije Universiteit Amsterdam, 5 July 2023, 18:45-21:30

You may use (a hard copy or print-out of) the textbook Distributed Algorithms: An Intuitive Approach. *Use of solutions to exercises, slides, notes, laptop is* not *allowed.*

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

1. Explain how vector clock values can be computed at run-time. (12 pts)

   Solution: VC can be computed at runtime as follows. Let a be an event at a process p_i, and $(k_0, \ldots, k_{N-1})$ the clock value of the previous event at p_i (take $(0, \ldots, 0)$ if there is no such previous event).

   ∗ Suppose a is an internal or send event. Then $VC(a) = (k_0, \ldots, k_i{+}1, \ldots, k_{N-1})$.

   ∗ Suppose a is a receive event. Let sender p_j attach the clock value $(\ell_0, \ldots, \ell_{N-1})$ of the corresponding send event to the message. Then $VC(a) =$

   $$(\max\{k_0, \ell_0\}, \ldots, \max\{k_{i-1}, \ell_{i-1}\},\ k_i{+}1,\ \max\{k_{i+1}, \ell_{i+1}\}, \ldots, \max\{k_{N-1}, \ell_{N-1}\}).$$

2. Give an example where in the Peterson-Kearns rollback recovery algorithm a process must retrieve from one of its checkpoints a basic message it needs to resend to the crashed process. (12 pts)

   Solution: Process q sends a message m to process p, takes a checkpoint, crashes, and recovers. Next, p crashes, and m reaches p between p's last checkpoint and its recovery. Now q must resend m, which it needs to retrieve from its checkpoint.

3. Give an example run of the Dolev-Klawe-Rodeh election algorithm on a directed ring of seven processes to show that an active process can receive a message for two rounds ahead. (12 pts)

   Solution: Consider seven consecutive active processes in the directed ring with IDs 1, 5, 2, 7, 3, 6, and 4, respectively. In round 0, process 4 sends a message to its successor 1, which is slow. Processes 2, 3, and 4, after receiving messages from their two nearest predecessors, remain active in round 1, where they assume ID 5, 7, and 6, respectively. Processes 7 and 6 become passive in round 0, while processes 1 and 5 keep waiting for a message from process 4 in round 0. In round 1, process 4 (now carrying ID 6), after receiving messages from its nearest two active neighbors in this round, sends a second message to its successor 1, which again is slow. Since process 4 receives IDs 5 and 7 in round 1 and carries ID 6, it remains active in round 2, where it assumes ID 7 and sends a third message to its successor 1. This message overtakes the earlier two messages that process 4 sent.

4. Propose an adaptation of the Itai-Rodeh election algorithm in which the Boolean fourth parameter in messages, to recognize whether another process selected the same random ID in the current election round, is omitted. (14 pts)

   Solution: Each active process p maintains a Boolean, which is reset at the start of each new election round, and which is set if a message for the current election round with the same ID as p's ID in this round arrives with a hop count smaller than N.

   If p receives its own message back and its Boolean is set, it starts the next election round, with a new random ID.

   If p receives its own message back and its Boolean is not set, it becomes the leader. From then on it will not pass on any more messages.

   When a process becomes the leader, the computation may not yet have been terminated. So the leader will have to announce that it has become the leader, as else other processes that selected the same ID as the leader in the last round may stay active forever.

5. Consider a complete network of five processes. Apply the Chandra-Toueg 2-crash consensus algorithm, where initially four processes choose the value 0 and one process the value 1. Give a computation in which all correct processes decide for 1. (12 pts)

   Solution: Initially, p_0 has value 1, while p_1, p_2, p_3, p_4 have value 0.

   In round 0, p_0 gets $\langle \mathbf{vote}, 0, 1, -1\rangle$ from itself and $\langle \mathbf{vote}, 0, 0, -1\rangle$ the other four processes. Since all these messages have the same *last-update* value -1, p_0 can pick any of the values it received, and arbitrarily picks its own value 1.

   p_0 broadcasts $\langle \mathbf{value}, 0, 1\rangle$

p_0 receives $\langle \textbf{ack}, 0 \rangle$ from all processes.

p_0 decides for 1, and broadcasts $\langle \textbf{decide}, 1 \rangle$.

All other processes receive this message, and also decide for 1.

6. Suppose that in a run of the BB84 quantum key exchange protocol, Eve checks all bits sent from Alice to Bob. How many mistakes is she expected to introduce in the secret key computed by Bob? (9 pts)

   Solution: On roughly $\frac{n}{2}$ of the n bits sent by Alice, Eve will guess correctly whether the Hadamard transform was applied by Alice. These bits will give the correct outcome at Bob (assuming that on bits to which Eve applied the Hadamard transform, she applies the Hadamard transform once again before passing them on to Bob).

   On the roughly $\frac{n}{2}$ bits where Eve guesses wrong, she has a 50% chance of introducing an error at Bob. So on roughly $\frac{n}{4}$ bits she introduces a mistake at Bob.

   On roughly half, so $\frac{n}{8}$, of these bits will Alice and Bob both have done the same thing (both applied the Hadamard transform or both didn't do so).

   Concluding, of the $\frac{n}{2}$ bits on which Alice and Bob both have done the same thing, Bob is expected to have read the wrong value on 25% of these bits, due to the interference of Eve.

7. Let Alice and Bob build a private key using the Diffie-Hellman key exchange protocol, with $p = 17$ and d the smallest positive integer that is a primitive root modulo 17. Moreover, let $a = 2$ and $b = 3$. Explain how Alice and Bob construct their private key. (10 pts)

   Solution: The smallest primitive root modulo 17 is 3, because its respective powers modulo 17 are: $3, 9, 10, 13, 5, 15, 11, 16, 14, 8, 7, 4, 12, 2, 6, 1$. (2 is not a primitive root: $2, 4, 8, 16, 15, 13, 9, 1$.)

   Alice computes $3^2 = 9 \bmod 17$ and sends 9 to Bob. Bob computes $3^3 = 27 = 10 \bmod 17$ and sends 10 to Alice.

   Alice computes $10^2 = 100 = 15 \bmod 17$ and Bob computes $9^3 = 729 = 15 \bmod 17$. So their private key is 15.

8. In the Winternitz signature, why would the checksum $b_1 + \cdots + b_n$ be less effective against replay attacks than the actual checksum? (9 pts)

   Solution: Let Eve try a replay attack with binary numbers $c_1, , c_n$, where $c_i \geq b_i$ for $i = 1, \ldots, n$. The checksum $c_1 + \cdots + c_n$ for attacker Eve produces a sequence of binary numbers $c_{n+1} \cdots c_m$ that may all be greater or equal than the sequence of binary numbers $b_{n+1} \cdots b_m$ produced by Alice's checksum. That is, possibly $c_i \geq b_i$ for $i = n+1, \ldots, m$. Thus Eve's replay attack might still be possible with regard to the extended signature that takes into account the checksum.