Resit Distributed Algorithms

Vrije Universiteit Amsterdam, 6 July 2022, 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. Propose an adaptation of the Lai-Yang snapshot algorithm in which basic messages may be buffered at the receiving processes and in which the channel states of the snapshot are always empty. (15 pts)

   Solution: Key is that a process must have received all basic messages that were sent to it presnapshot before computing its local snapshot. This avoids that such messages must be included in the incoming channel state.

   An initiator, or a noninitiator that receives a control message or a basic message with *true* for the first time, sends a control message into each outgoing channel, informing the process at the other side how many presnapshot basic messages were sent into it. It stores its local state, in order to later compute its local snapshot. And it starts appending *true* to the basic messages it sends.

   Presnaphot basic messages, without *true* attached, that are received through an incoming channel are buffered until the receiver has taken its local snapshot.

   When a process has received all presnapshot basic messages through all its incoming channels, it computes its local snapshot, by performing the buffered basic messages as if they were all received right after the stored local state.

2. In the Bracha-Toueg deadlock detection algorithm, can also noninitiators, after having received a **done** from the neigbors to which they sent a **notify**, see from their *free* field whether they are deadlocked? Explain your answer. (10 pts)

   Solution: Yes, because each deadlock detection run cleans out the subtree of the wait-for-graph at all nodes participating in the run as much as possible.

   (This only applies to nodes that take part in the deadlock detection run. Here it is important that the initiator sends **grant** messages.)

3. Suppose that in Rana's termination detection algorithm, processes can take part in a wave tagged with t if they have been quiet from some time $\leq t+5$. Give an example to show that then termination could be detected prematurely. (10 pts)

Solution: Suppose quiet process p starts a wave tagged with t, setting its logical clock to $t+1$. Active process q sends a basic message to p, the receipt of which makes p active and pushes its logical time to $t+2$. p sends back an ack to q, with logical time stamp $t+3$, which upon arrival pushes q's logical clock to $t+4$. Now q becomes passive, setting its logical clock to $t+5$, and as a result becomes quiet. When the wave finally arrives, q participates, and the wave may complete, although p is active.

4. Give a Monte Carlo algorithm for election in anonymous networks of unknown size, and analyze the success probability of your algorithm. (15 pts)

Solution: Suppose there are N initiators.

The simplest solution is to let each initiatior decide with some probability $\pi \in \langle 0,1\rangle$ that it is the leader. The probability that a given initiator becomes the unique leader is $\pi \cdot (1-\pi)^{N-1}$: the given initiator becomes the leader with probability π, the other $N-1$ initiator do not become the leader with probability $1-\pi$. Since each of the N initiators can be the one to become the leader, the overall success probability is $N \cdot \pi \cdot (1-\pi)^{N-1}$.

Another solution is to use the echo algorithm with extinction for one election round, whereby each initiatior selects a random ID from $\{1,\ldots,R\}$. This algorithm terminates correctly, i.e., with one leader, if there is one initiator with an ID larger than all other initiators. The probability that this is the case is:

$$\sum_{i=1}^{R} N \cdot \left(\frac{i-1}{R}\right)^{N-1}$$

The $(i-1)^{N-1}$ in the numerator expresses for each ID i the total number of initial configurations in which all other initiators choose an ID smaller than i: these $N-1$ initiators can choose from $i-1$ IDs. The R^{N-1} in the denominator is the total number of initial configurations for the other initiators: these $N-1$ initiators can choose R different IDs.

5. Explain how the Ricart-Agrawal mutual exclusion algorithm could be adapted to make it fault-tolerant. (15 pts)

Solution: Let us assume a (complete and) strongly accurate failure detector.

If a process receives a request, it checks whether maybe the sender has crashed. If so, the request is purged.

If a process finds that another process crashed, it purges pending requests to and from that process, and records that it no longer needs permission from the crashed process to become privileged.

6. Sketch how the AODV protocol can be adapted to allow a node to look for multiple minimum-hop paths to different destinations with the broadcast of a single RREQ message. (15 pts)

Solution: Suppose node p looks for paths to destinations $q_1, \ldots, q_n$ by broadcasting a single RREQ message. First of all, the IDs of the destinations need to all be included in the RREQ messages.

Let an RREQ arrive at a node r with hop count h, where r does not have an active route to p with a more recent sequence number than sn_p, or with the sequence number sn_p and a distance value $d \leq h$. If r does not have an active route to one or more destinations in the RREQ, then it broadcasts the currently received RREQ, with the hop count increased by 1, including only those destinations in the RREQ to which it does not have an active route. If on the other hand r does know an active route to one or more destinations $q_{i_1}, \ldots, q_{i_k}$ in the RREQ, then it answers with an RREP, which contains the IDs of p and $q_{i_1}, \ldots, q_{i_k}$, the overall distance of each route to the k destinations, and the sequence number of each route, originating from $q_{i_1}, \ldots, q_{i_k}$. In particular, if $r = q_i$ for some i, then the RREP is provided with the sequence number of r, which is then increased by 1. Suppose a node s receives such an RREP. For each destination q_i that in the message carries a more recent sequence number than s's current route to q_i, or the same sequence number and yielding a shorter route to q_i, s updates its routing information to q_i. If s is not p and updated its routing information by the received RREP, then s forwards this RREP toward p, preserving only those (one or more) destinations q_i's, and the corresponding information (overall distance of the route, sequence number), for which s updated its routing information.

7. Suppose a one-time signature *sig* is placed in the fourth leaf of a binary Merkle tree of depth 4 and used by Alice in a Merkle signature of a message to Bob. Explain what the signature looks like and how this signature is employed by Bob to verify whether the public key is genuine. (10 pts)

Solution: Alice's signature takes the form $sig_4 \,\|\, Y_4 \,\|\, h(Y_3) \,\|\, H_{31} \,\|\, H_{22} \,\|\, H_{12}$ with sig_4 the signature and Y_3, Y_4 the third and fourth public key. Each H-value 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_4 to determine the value in the fourth 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.