Distributed Systems and Algorithms— CSCI 4510/6510
Final Exam

Due: Wednesday, Dec. 12, 2018 at 10:00pm
Optional no-penalty extension until Friday, Dec. 14, 2018 at 11:00pm
8 questions (100 points)

1. **[15 points]** In class, we learned several distributed mutual exclusion algorithms that can be used to guarantee one-at-a-time access to a resource. In the k mutual exclusion problem, the objective is to guarantee that at most $k \geq 1$ processes can access the resource at any given time.

   (a) Select a mutual exclusion algorithm that we studied in class, and describe how it can be extended to solve the k mutual exclusion problem.

   (b) Identify the system model for your extended algorithm and state whether your algorithm guarantees safety and liveness in this system model.

   (c) Assume k processes request the resource simultaneously. What is the worst-case message complexity of your algorithm in this case? Include messages required to request, access, and release the resource (when applicable).

2. **[15 points]** In class, we studied the Chandy-Lamport distributed snapshot algorithm for recording a consistent global state. A similar concept to a consistent global state is a *transitless global state*. A transitless global state is one in which every message that has been sent has also been received.

   Dr. Science proposes the following distributed snapshot algorithm. Let $T > 0$ be an integer known to all processes. Every process maintains a Lamport clock (initialized to 0) that it uses to timestamp all events that take place on that process. Each process p_j keeps track of the timestamps of incoming messages. Once p_j has received a message with timestamp greater than or equal to T from every other process, p_j records its state. No channel states are included in the snapshot.

   Assume that each process communicates with each other process infinitely often and that the channels are asynchronous, reliable, and FIFO.

   (a) Does Dr. Science's algorithm record a consistent global state? Justify your answer.

   (b) Does Dr. Science's algorithm record a transitless global state? Justify your answer,

3. **[10 points]** An algorithm that solves the Two Generals Problem must satisfy the following properties: agreement, validity, and termination. In class, we learned that in a system with two nodes (generals) that never fail, but where messages may be lost, there is no algorithm that solves the Two Generals Problem. Suppose that we strengthen the communication model so that it is guaranteed that at most three messages will be lost in total, and both generals are aware of the communication model. Is it possible to solve the Two Generals Problem under this model? If so, give an algorithm that solves the problem, and explain why it solves the problem. If not, explain why not?

4. **[10 points]** Recall the Three-Phase Commit algorithm for synchronous systems with crash failures.

   (a) Describe an execution of the algorithm in which all participants (including the coordinator) vote commit, but the active processes decide abort.

   (b) Explain how this execution satisfies the agreement, validity, and termination properties.

5. **[15 points]** Recall the Diffusion Algorithm for reliable broadcast in a system with asynchronous messaging and crash failures. Suppose that channels are also FIFO. Does the Diffusion Algorithm implement FIFO Reliable Broadcast in this system? Specifically, does it guarantee the properties: validity, agreement, integrity, and FIFO order? Justify your answer.

6. **[15 points]** Describe how the Paxos algorithm can be used to implement Atomic Broadcast. Explain how, and under what system model, your implementation satisfies the properties: validity, integrity, agreement, and total order.

7. **[10 points]** Let a *set* be a data structure that supports two operations:

   - *add(d)*: insert data object d into the set.
   - *remove()*: if the set is not empty, remove one object from the set and return it. Else, return *null*.

   Consider a system where a set s is replicated at all Replica Managers. Assume the system has asynchronous reliable messaging and no failures. The system uses a quorum-based approach to implement sequentially consistent replication of s. Let $V(s)$ be the total voting weight of the system for the set s. To execute the *add* operation, a Front End server needs to access an *add quorum*, and to execute the *remove* operation, a Front End server needs to access a *remove quorum*. Assume that $add(d)$ is called at most once for each object d.

   Develop and describe an implementation of the *add* and *remove* operations in this system that guarantees sequential consistency. Be sure to state which parts of these operations must be performed atomically, if any. Also state the *add threshold* $A(s)$ and the *remove threshold* $R(s)$ for your implementation. To receive full credit for this question, at least one of your thresholds must be strictly less than $V(s)$.

   Hint: You may want to store the set as a collection of *add* and *remove* operations rather than as a collection of data objects.

8. **[10 points]** A *range query* is a database operation that returns all data items whose value is between a specified lower and upper bound. An example of a range query is $range(\mathrm{A}, \mathrm{C})$, which should return all files with names that start with the letter 'A', the letter 'B', or the letter 'C'.

   (a) Suppose you are asked to implement an extension to the Chord DHT to support range queries. Briefly describe how you would do this.

   (b) What is the message complexity of a single range query in your implementation?

   (c) Is your implementation guaranteed to retrieve all files in the query range even when there is churn? If so, explain why. If not, describe an example execution where the response to a query does not contain all files in the specified range.