

# Instructions

For the main questions, be ready to:

- Describe the architecture of your solution and its operations.
- Define all its components and their interactions.
- Motivate your design decisions.
- Make diagrams whenever necessary.
- State clearly your choices and assumptions

For feedback, feel free to e-mail [joeri.hermans@doct.uliege.be](mailto:joeri.hermans@doct.uliege.be)

## Question 1.

You are responsible for a peer-to-peer messaging system (no centralization). It is your responsibility that a message sent by a user is eventually delivered to all users currently present in the messaging room. What assumptions do you make on the ordering of the messages? What broadcast abstraction should be used here? Why? Why are Best-Effort Broadcast and Reliable Broadcast not sufficient?

Discuss a specific implementation in the *fail-stop* distributed system model.

**Discuss:** What (link) failure detector is present? Discuss its implementation.

## Question 2.

Discuss the broadcast topology in relation with the number of messages transmitted. Is the conventional centralized architecture costly? Are there alternatives? What are the trade-offs?

## Question 3.

You are tasked with the design of a low-power mesh network with an arbitrary number of participants. Mesh network members can arbitrarily leave and join the network. What broadcast or messaging primitives would you use to distribute messages through the network? Implement a mechanism which prevents that messages are *delivered* twice. Give several examples of possible executions of message dissipation for various network architectures. Is network

architecture an important aspect? Discuss its importance in relation to the number of messages and noise of the channel?

**Extension:** Is your approach resilient to deal with a *stream* of messages? If yes, why? If not, how would you restructure your architecture?