Chalmers University of Technology

Computer Science and Engineering
Elad Michael Schiller

Written exam in EDA387/DIT670 Computer Networks 2022-08-25. Exam time: 4 hours.

Examiner: Elad Michael Schiller, phone: 073-6439754

Credits:	30-38	39-47	48-Max
Grade:	3	4	5
Grade (GU)	G	G	VG

1. The answer must be written in English (even for Swedish-speaking students). Use proper grammar and punctuation.
2. All answers need to be motivated unless otherwise stated. Correct answers without motivation or with wrong motivation will not be given full credit.
3. Answer concisely, but explain all reasoning. Draw figures and diagrams when appropriate.
4. Write clearly. Unreadable or hard-to-read handwriting will not be given any credit.
5. Do not use red ink.
6. Solve only one problem per page.
7. Sort and number pages by ascending problem order.
8. Anything written on the back of the pages will be ignored.
9. Do not hand in empty pages or multiple solutions to the same problem. Clearly cross out anything written that is not part of the solution.
10. Do not hand in anything that is not part of the solution.
11. Always write your answers in your own words, using your own diagram, etc. In case you have to use the words of others, e.g., a formal definition, please make sure to use quotation marks and clearly identify the source.
12. Absolutely do not collaborate in any way or form with other people during the entire exam time (regardless of whether you are still writing the exam or not).

Question 1: General Questions (17 points in total)

1.a (3 points) In class we learned about self-stabilizing algorithms for constructing a distributed spanning tree. **(1 pt)** What rule does this algorithm serve in the construction of routing tables over the Internet? **(1 pt)** Please name the exact self-stabilizing algorithm and the routing protocol. **(1 pt)** Please explain how the algorithm is used and what for.

1.b (4 points) This part focuses on software-defined networks (SDNs). **(1 pt)** Explain what is the functionality that packet forwarding rules serve in SDNs. **(3 pt)** Do any of the algorithms considered in item 1.a can help to satisfy the task of constructing in-band routing in SDNs? If not, please explain why. If it can help, please provide detail regarding how it can be done.

1.c (2 points) IPv4 address has 32-bits whereas IPv6 address has 128-bits. **(1 pt)** Please explain the motivation for this change. **(1 pt)** Please clarify the limitations of the Internet without this change.

1.d (4 points) Suppose that the Internet address had 74819-bits already in version 1. **(1 pt)** In your opinion, how would the Internet have developed? **(1 pt)** What limitations the Internet would have had and what limitation it would not have had. **(1 pt)** Are there any new protocols that would be needed? **(1 pt)** Are there any existing protocols that would not be needed? Please specify these protocols, their names, and their functionality.

1.d (4 points) (2 pt) In what sense a solution for super-self-stabilizing systems is better than a self-stabilizing one? **(2 pt)** Should we just focus on super-self-stabilization instead of self-stabilization?

.....

Question 2: Selecting the right phrases from a list of phrases (3 points in total)

Three phrases are missing in Fig 1. Please complete the missing phrases using only three phrases from the following list of nine phrases. (a) timeouts, (b) slow start period, (c) selective acknowledgment, (d) probing for usable bandwidth, (e) two duplicate ACKs, (f) fast retransmission, (g) maximum segment size, (h) explicit congestion notification, and (i) additive increase. Choose only the most appropriate phrases. In case you are not sure about your answer, you are welcome to explain yourself briefly (but you don't have to).

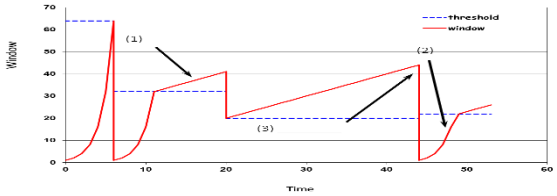

Figure 1 select the correct phrases

.....

Question 3: Non-self-stabilizing digital clock synchronization (6 points)

3a. (1 point) Please define the task of digital clock synchronization by filling up the following sentences.
[agreement]: _____ clock values
[progress]: the clock values are _____ every pulse

Fig. 4 contains a non-self-stabilizing algorithm for digital clock synchronization.

```
01 upon a pulse
02     forall P_j ∈ N(i) do send (j,clock_i)
03     let lClocks := ∅
04     forall P_j ∈ N(i) do
05         receive(clock_j)
06         lClocks := lClocks ∪ {clock_j}
07     od
08     clock_i := 1+ max(lClocks)
```

Figure 2 Non-self-stabilizing digital clock synchronization

3b. (1 point) Please explain why this algorithm is not self-stabilizing.

3c. (4 points) Please demonstrate that within $O(d)$ pulses, a system that run the algorithm presented in Fig. 4 satisfies the task of digital clock synchronization, where d is the graph diameter. Your proof should include all the key steps of the complete proof. Please assume that transient faults cannot occur (not even before the system starts running). Moreover, please use the assumption that were used in class.

Question 4: Self-stabilizing maximal independent set in general graphs with a distinguished node (14 points)

This question considers the design of a deterministic self-stabilizing algorithm for maximal independent set construction in a network that its topology is a general graph with a distinguished node.

Definitions

An independent set in graph G = (V, E) is a set S ⊆ V of vertices, such that no two of members of S are adjacent in graph G. That is, for every two vertices in S, there is no edge connecting the two. Equivalently, each edge in E has at most one endpoint in S. A maximal independent set (MIS) is an independent set that is not a subset of any other independent set. In other words, there is no vertex outside the independent set that may join it because it is maximal with respect to the independent set property.

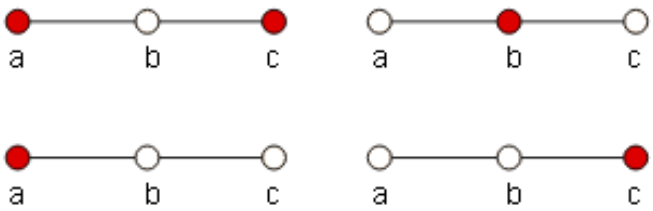

Figure 1 The top two graphs are maximal independent sets while the bottom two are independent sets, but not maximal. The maximum independent set is represented by the top left.

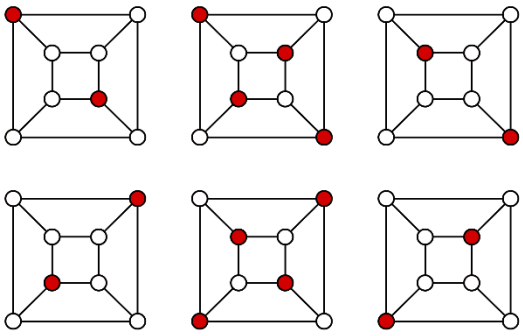

Figure 2 The graph of the cube has six different maximal independent sets (two of them are maximum), shown as the dark

We say that a non-deterministic execution of a self-stabilizing algorithm follows the central demon model when at any time at most one processor may execute its do forever loop, i.e., be in the middle of an incomplete do forever loop. We say that a non-deterministic execution of a self-stabilizing algorithm follows the distributed demon model when more than one processor at a time can execute concurrently its do forever loop.

4a (5 points) Suppose you are given a deterministic self-stabilizing algorithm for maximal independent set construction in a network that its topology is a general graph that assumes the use of a central demon and has no distinguished node or unique node identifiers. However, the network that you need to solve the problem for assumes a distributed demon model, but it has a distinguished node. Please describe how the algorithms and techniques used in class, as well as the ones provided by the book, that can help to implement the given algorithm for maximal independent set construction over the network settings given above. You are welcome to draw figures and use examples for clarifying your answer. In other words, we need to covert the given algorithm to another one that can run over the given network.

4b (5 points) Please give a clearly written pseudo-code for a deterministic self-stabilizing algorithm for maximal independent set construction in a network that its topology is a general graph under the assumption of a central demon model.

4c (2 points) Please define the set of safe configurations (legitimate system states) for your self-stabilizing algorithm for maximal independent set construction and your proposed solution.

4d (2 points) Please provide a correctness proof of the closure property for your solution in 4b. Please bring all the arguments needed to convince the reader that your proposal never leaves the set of legal executions.

Question 5: Token circulation in a directed ring with a distinguished node (10 points)

5a. (2 points) Define the task of token circulation.
5b. (4 points) Show that, without a distinguished processor, there is no deterministic solution of the token circulation.

Below (in Figure 3) please find Dijkstra algorithm for token circulation in directed rings as well as the proof.

```
01  P1:          do forever
02                    if x1=xn then
03                          x1:=(x1+1)mod(n+1)
04  Pi(i ≠ 1):   do forever
05                    if xi ≠ xi-1 then
06                          xi:=xi-1
```

Figure 3 Dijkstra algorithm for token circulation

- A configuration in which all x variables are equal, is a safe configuration for ME (Lemma 2.2)
- For every configuration there exists at least one integer j, such that for every Pi, the value of xi is not equal to j (Lemma 2.3)
- For every configuration c, in every fair execution that starts in c, P1 changes the value of x1 at least once in every n rounds (Lemma 2.4)
- For every possible configuration c, every fair execution that starts in c reaches a safe configuration with relation to ME within O(n2) rounds (Theorem 2.1)

5c. (4 points) Both the proof of Lemma 2.2 and 2.4 make arguments about the prorogation of x_1 within n-1 asynchronous cycles. Please compare the way that the proof of these two lemmas uses these arguments. What is similar? What is different?

Question 6: Self-stabilizing maximum matching on a directed ring with a distinguished node (10 points)

Let $p_0, \ldots, p_{n-1}$ be *n* processors on a directed ring that Dijkstra considered in his self-stabilizing token circulation algorithm (in which processors are semi-uniform, i.e., there is one distinguished processor, p_0, that runs a different program than all other processors, but processors have no unique identifiers). Recall that in Dijkstra's directed ring, each processor p_i can only read from $p_{i-1 \bmod n}$'s shared variables and each processor can use shared variables of constant size. Design a deterministic self-stabilizing matching algorithm that always provides an optimal solution.

6a. (4 points) Please give a clearly written pseudo-code.
6b. (1 points) Please define the set of legal executions.
6c. (4 points) Please provide a correctness proof with all the arguments needed to convince the reader that the algorithm is correct and self-stabilizing.
5d. (1 points) Show that your algorithm is always optimal after convergence. What is the number of states (in the finite state machine that represents each processor) that your algorithm needs? Prove your claims.