Chalmers University of Technology
Computer Science and Engineering
Elad Michael Schiller

2015-10-30

Written exam in EDA387/DIT663 Computer Networks 2015-10-30. Exam time: 4 hours.

Means allowed: Nothing except paper, pencil, pen and English - xx dictionary.

Examiner: Elad Michael Schiller, phone: 073-6439754
Note that student questions can be answered only by phone.

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 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.

Svar på PingPong

Question 1: DNS Lab (6 points)

A user issued the following **dig**-command to find specific DNS information. The output of running the command is shown below.

```
C:\dig> dig mx ntnu.no @ns1.ntnu.no
```

```
; <<>> DiG 9.3.2 <<>> mx ntnu.no @ns1.ntnu.no
;  (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1780
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; QUESTION SECTION:
;ntnu.no.                        IN        MX

;; ANSWER SECTION:
ntnu.no.               60        IN        MX        10 mx.ntnu.no.

;; AUTHORITY SECTION:
ntnu.no.               3600      IN        NS        ns2.ntnu.no.
ntnu.no.               3600      IN        NS        ns1.ntnu.no.

;; ADDITIONAL SECTION:
mx.ntnu.no.            600       IN        A         129.241.56.67
ns1.ntnu.no.           3600      IN        A         129.241.0.208
ns2.ntnu.no.           3600      IN        A         129.241.0.209

;; Query time: 32 msec
;; SERVER: 129.241.0.208#53(129.241.0.208)
;; WHEN: Mon Aug 17 11:52:11 2015
;; MSG SIZE  rcvd: 240
```

When answering the following sub-questions, please note that it is NOT enough with copy from the above output. Use DNS-terminology.

1a. (1 point) What did the user want to ask about? Explain by referring to the syntax of the command.

1b. (1 point) Explain the contents of the "QUESTION SECTION". What is the name of the object type that was queried?

1c. (2 points) To which DNS server (hostname and IP-address) was the query sent? Is it an authoritative server or not? Explain and point out how you are able to confirm that it is or not.

1d. (2 points) Is there any answer in the reply? Describe in detail the information that the reply is providing to the user?

Question 2: DNS (6 points)

Suppose that you are using the Chalmers network to connect your laptop to the Internet. Suppose also that you want to access the website www.tue.nl for the first time (not in cache). Explain **how and why** DNS will be involved immediately after entering the name of the site in your browser. The answer should, specifically and technically, explain the necessary operation, including:

- the interaction and communication between the different DNS resolvers and servers,
- the protocols and messages used, and
- the final outcome.

Please remember that you should use **DNS terminology.**

Question 3: IPv6 Addresses and Neighbor Discovery (10 points)

These three IPv6 addresses are given with the compressed representation:

(i) 2001:6b0:2:10::1 uni/any
(ii) FF02::1:ff6c:14dd multi
(iii)FE80::20c:f1ff:fe6c:14dd uni

Note: Please answer the following sub-questions (3a, 3b, and 3c) separately and in relation to the above addresses. In addition, you are allowed to use any of the given addresses as examples in your answers to sub-questions 3d and 3e.

3a. (2 points) What is the "type" of each of these IPv6 addresses? Explain what each type does imply.

3b. (1 point) Which of the given addresses **cannot** be used as valid source address in IPv6 packet? Explain why. network boundary

3c. (2 points) What is the "scope" of each of these IPv6 addresses? Explain what each scope does imply.

3d. (2 points) What is the **main** purpose of IPv6 Neighbor Discovery? Explain **clearly** the operation.

3e. (3 points) What are the messages deployed in IPv6 Neighbor Discovery? Explain how these messages will be encapsulated and addressed in layer-3 and layer-2 PDUs (i.e. packets and frames).

Question 4: Dijkstra's self-stabilizing algorithm for token circulation (4 points)

Please find below Dijkstra's self-stabilizing algorithm for token circulation, as well as the proof outline, see Lemma 2.2 to 2.4 and Theorem 2.1. Please prove one of these proof elements, i.e., either Lemma 2.2, 2.3, 2.4 or Theorem 2.1.

```
P_1:           do forever
                   if x_1=x_n then
                       x_1:=(x_1+1)mod(n+1)
P_i(i ≠ 1):    do forever
                   if x_i ≠ x_{i-1} then
                       x_i:=x_{i-1}
```

- A configuration in which all x variables are equal is a safe configuration for the set of legal execution ME (Lemma 2.2)
- For every configuration there exists at least one integer j such that for every p_i, x_i is not equal to j (Lemma 2.3)
- For every configuration c, in every fair execution that starts in configuration *c*, p_1 changes the value of x_1 at least once in every n rounds (Lemma 2.4)
- For every possible configuration c, every fair execution that starts in configuration *c* reaches a safe configuration with relation to ME within $O(n^2)$ rounds (Theorem 2.1)

Question 5: self-stabilization leader election (18 points)

5a. (1 point) Define the set of legal executions for leader election for general communication networks.

Below please find a non-self-stabilizing leader election algorithm for general networks in which processors have access to unique identifiers.

```
initialization
    pulse_i := 0
    ⟨candidate,distance⟩ := ⟨ ID(i), 0 ⟩
while pulse_i ≤ D do
    forall P_j ∈ N(i) do
      begin
          ⟨leader_i[j],dis_i[j]⟩ := read⟨ leader_j, dis_j ⟩
          if ((leader_i[j] < candidate) or
             ((leader_i[j] = candidate) and (dis_i[j] < distance)) then
                 ⟨candidate,distance⟩ := ⟨ leader_i[j],dis_i[j] + 1⟩
      end
    write⟨leader_i,dis_i⟩ := ⟨candidate,distance⟩
```

5b. (2 points) Suppose that after the execution of line 03 by all processors, a transient fault occurs that change the state of a single processor. As a result, the system elects a leader that does not exist in the system, i.e., a floating identifier has been introduced to the system during the transient fault. Please define an example of a system configuration after that transient fault leads to the above result of the algorithm run.

The rest of this question is whether there exists a deterministic self-stabilizing leader election algorithm for general networks in which processors have access to unique identifiers.

Please answer either sub-question 5c.1 or 5c.2 (not both).

5c.1 (15 points) If you think that there is no such self-stabilizing algorithm, please prove your claim.

5c.2 If you think that there is such self-stabilizing algorithm, please design a deterministic self-stabilizing algorithm for leader election in general networks in which processors have access to unique identifiers.

> **5c.2a (5 points)** Write the pseudocode of your algorithm.
>
> **5c.2b (2 points)** Define a safe configuration with respect to the set of legal executions (that you have defined) and the algorithm (that you have given the pseudo-code for).
>
> **5c.2c (6 points)** Demonstrate that your algorithm can overcome the floating identifier problem.
>
> **5c.2d (2 points)** Give the idea of the entire convergence property of your algorithm.

Question 6: self-stabilization vertex coloring (12 points)

6a. (1 points) Define the set of legal executions for the task of vertex coloring in general graphs.

Let $p_1, \ldots, p_n$ 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 processor, p_1, that is different than other processors but there are no unique identifiers). Consider only deterministic algorithms and the task of vertex coloring in Dijkstra's directed ring. The rest of this question is whether there exists a deterministic self-stabilizing algorithm for coloring these networks.

Please answer either sub-question 6b.1 or 6b.2 (not both).

6b.1 (11 points) If you think that there is no such self-stabilizing algorithm, please prove your claim.

6b.2 If you think that there is such self-stabilizing algorithm, please design a deterministic self-stabilizing algorithm for vertex coloring for Dijkstra's directed ring.

> **6b.2a (2 points)** Write the pseudocode of your algorithm.
>
> **6b.2b (1 point)** Define a safe configuration with respect to the set of legal executions (that you have defined) and the algorithm (that you have given the pseudo-code for).
>
> **6b.2c (2 points)** Demonstrate the closure property.
>
> **6b.2d (4 points)** Demonstrate the convergence property.

25

6b.2d (2 points) Suppose that the number of processors, n, is an even number. Does your algorithm uses an optimal number of colors? Prove your claims. What is the minimal number of colors needed for vertex coloring Dijkstra's directed ring? How many colors does your algorithm use?

Question 7: self-stabilization (digital) clock synchronization (12 points)

7a. (2 points) Define the set of legal executions for (digital) clock synchronization.
(1) The property of *agreement* means that ________ (complete this sentence).
(2) The property of *(no) adjustment* means that ________ (complete this sentence).

Let $p_1, \ldots, p_n$ be n processors on a uniform directed ring (in which there are no processors identifiers and all processors run the same program). Consider only deterministic algorithms and the task of (digital) clock synchronization in uniform directed ring. The rest of this question is whether there exists a deterministic and uniform self-stabilizing algorithm for (digital) clock synchronization for these networks ***that uses only a constant number of states per processor***.

Please answer either sub-question 7b.1 or 7b.2 (not both).

7b.1 (10 points) If you think that there is no such self-stabilizing algorithm, please prove your claim.

7b.2 If you think that there is such self-stabilizing algorithm, please design a deterministic self-stabilizing algorithm for (digital) clock synchronization for uniform directed rings.

7b.2a (2 points) Write the pseudocode of your algorithm.

7b.2b (2 points) Define a safe configuration with respect to the set of legal executions (that you have defined) and the algorithm (that you have given the pseudo-code for).

7b.2c (2 points) Demonstrate the closure property.

7b.2d (4 points) Demonstrate the convergence property.