Chalmers University of Technology
Computer Science and Engineering
Elad Michael Schiller

2013-10-25

Written exam in EDA387/DIT663 Computer Networks 2013-10-25. Exam time: 4 hours.

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

Examiner: Elad Michael Schiller, phone: 073-6439754 and 031-7721052

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.

Question 1 DNS (12 points)

dig (domain information groper) is a useful command-line tool for querying the name system of the Internet. This tool has been used extensively during one of the course labs.

A PC-user issues the **<dig>** command in order to get DNS-information. Examine the output carefully and then answer the questions given below (appears on next page) using DNS-terminology and concepts. Please answer each question separately.

C:\dig>dig mx chalmers.se @ns1.chalmers.se

```
; <<>> DiG 9.3.2 <<>> mx chalmers.se @ns1.chalmers.se
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 670
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 7

;; QUESTION SECTION:
;chalmers.se.                  IN    MX

;; ANSWER SECTION:
chalmers.se.          14400   IN    MX    10 putnam.ita.chalmers.se.
chalmers.se.          14400   IN    MX    10 clea.ita.chalmers.se.

;; AUTHORITY SECTION:
chalmers.se.          172800  IN    NS    ns1.chalmers.se.
chalmers.se.          172800  IN    NS    ns2.chalmers.se.
chalmers.se.          172800  IN    NS    dns.uu.se.
chalmers.se.          172800  IN    NS    ns3.chalmers.se.

;; ADDITIONAL SECTION:
clea.ita.chalmers.se.    14400   IN    A       129.16.222.61
putnam.ita.chalmers.se.  14400   IN    A       129.16.222.146
ns1.chalmers.se.         172800  IN    A       129.16.2.40
ns1.chalmers.se.         172800  IN    AAAA    2001:6b0:2:10::1
ns2.chalmers.se.         172800  IN    A       129.16.253.252
ns2.chalmers.se.         172800  IN    AAAA    2001:6b0:2:20::1
ns3.chalmers.se.         172800  IN    A       192.36.120.11

;; Query time: 120 msec
;; SERVER: 129.16.2.40#53(129.16.2.40)
;; WHEN: Thurs. Oct 10 20:19:56 2013
```

;; MSG SIZE rcvd: 288

Question 1 *Continued*

<u>**Note:**</u> Please answer the following questions with the aid of the issued command and syntax, and the information given in each section of the answer.

(1p) 1a. What DNS-information does the user specifically want to know? Relate the answer to the issued command.

(1p) 1b. To which DNS-server is the query sent? Give the complete hostname of this server.

(1p) 1c. Explain whether the user gets exactly the queried information or not. <u>***Note:***</u> NOT enough with answering "YES" or "NO".

(2p) 1d. How can you verify whether the above answer is authoritative or non-authoritative? Explain clearly in two different ways.

(2p) 1e. How many Resource Records are there in the answer section? Describe clearly the information and the content of **one** of these RRs.

(1p) 1f. Within each raw in the output, there is a given number (e.g. 14400 or 172800), explain what this number is telling and what it is used for.

(2p) 1g. What are the shown abbreviations (MX, NS, A, AAAA) standing for? Explain the meaning of each.

(1p) 1h. What is the IP address of the answering DNS-server?

(1p) 1i. Which protocol is used to transport DNS messages (the query and the answer) and what is the port number of the answering DNS-server?

Question 2 IPv6 Addresses (4 points)

(1p) 2a. Explain what is meant by the "scope" of an IPv6 address used as destination address in IPv6 packets.

(3p) 2b. What is the "type" and the "scope" of addresses which are configured out of each of the following IPv6 prefixes:

2001:6b0:2:10::   FF02::   FE80::

Question 3 IPv6 Autoconfiguration (6 points)

An IPv6 node will be connected to the Internet through an Ethernet-based network. The network has an attached router-interface which is periodically advertising the prefix 2001:06b0:2:10::/64. The node is configured to use stateless autoconfiguration of the interface's IPv6 addresses. Assume that the node's interface identifier is a288:b4ff:fe5c:c774.

(1p) 3a. When rebooting, what IPv6-address will be automatically configured for the interface in order to be used in Neighbor Discovery? When answering, please write the address in hexadecimal notation and give the type and scope of its use.

(1p) 3b. After rebooting, what IPv6-address will be automatically configured for the interface in order to be used for accessing the Internet? When answering, please write the address in hexadecimal notation and give the type and scope of its use.

(4p) 3c. Since there is neither broadcasting nor ARP in IPv6, describe the substituting operation, its purpose, the protocols and the messages used by this IPv6 node and its neighbors.

Question 4 (6 points)

Below please find an algorithm for digital clock synchronization. Namely, eventually it holds that we have: (1) identical clock values, and (2) the clock values are incremented by one, once in every pulse.

```
01 upon a pulse
02      forall P_j ∈ N(i) do send (j,clock_i)
03      max := clock_i
04      forall P_j ∈ N(i) do
05              receive(clock_j)
06              if clock_j > max then max := clock_j
07      od
08      clock_i := max + 1
```

4.a (2 p) The algorithm does not consider bounded number of values for the variable *clock*. Explain why in the context of self-stabilization this property is considered unattractive.

4.b (4 p) Is there any deterministic self-stabilizing algorithm (with bounded clock counter values)? If the answer is no, please give a formal impossibility proof. If the answer is yes, please write the code of the algorithm (or just say which line to change and in what why). Moreover, explain why the algorithm converges from any starting configuration (using between eight and eighteen sentences).

Question 5 (5 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 them, i.e., either Lemma 2.2, 2.3, 2.4 or Theorem 2.1, but <u>*just one of them*</u>!

```
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 ME (Lemma 2.2)

For every configuration there exists at least one integer j, such that for every processor p_i, the variable x_i is not equal to j (Lemma 2.3)

For every configuration c, in every fair execution that starts in 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 c reaches a safe configuration with relation to ME within $O(n^2)$ rounds (Theorem 2.1)

Question 6 (2 point)

6.a (1 p) Please complete the following sentence: "We say that ___ *c* is ___ with respect to the set of ___ ___, *LE*, if every system execution that starts from *c* is ___ *LE*."

6.b (1 p) Please define the set of legal executions, LE_{leader}, for the task of leader election. Please start by saying "We say that execution R is in LE_{leader} if for every .... it holds that .... "

Question 7 (8 points)

Please find below a self-stabilizing algorithm for leader election.

7.a (3 p) Please define the safe configuration of the algorithm. Make sure that you consider all variables and shared registers.

7.b (5 p). Suppose the system execution, *R*, starts in a safe configuration, *c*. Let a_i be a step that processor *pi* takes immediately after *c* and just before c'. Please show that c' is safe.

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

Question 8 (6 points)

Take a look at the self-stabilizing maximal matching algorithm. We assume the existence of a central daemon. Given a configuration *c*, we say that a processor p_i is:

- **matched** in *c*, if p_i has a neighbor p_j, such that $pointer_i = j$ and $pointer_j = i$.
- **single** in *c*, if $pointer_i = null$ and every neighbor of p_i is matched.
- **waiting** in *c*, if p_i, has a neighbor p_j such that $pointer_i = j$ and $pointer_j = null$.
- **free** in *c*, if $pointer_i = null$ and there exists a neighbor p_j, such that p_j is not matched.
- **chaining** in *c*, if there exists a neighbor p_j for which $pointer_i = j$ and $pointer_j = k$, $k \neq i$.

We define the variant function *VF(c)* as one that returns a vector (*matched + single, waiting, free, chaining*).

8.a (2 p) Please use the value of *VF(c)* to define the safe configuration, c. Hint: it is a vector that includes values that are either *0* or *n*. For that value show that: (1) $pointer_i = j$ implies that $pointer_j = i$, and (2) if $pointer_i = null$ then there is no neighbor p_j, such that if $pointer_j = null$.

8.b (4 p) Suppose that *c* is safe. Let a_i be a step of processor p_i that is taken immediately after *c*. Moreover, let *c'* be configuration that immediately follows by a_i. Show that c' is safe, i.e., the closure property. Hint: consider the case that a_i includes the execution of either line 02, 03 or 04. For each of these three cases, show that *VF(c)* = *VF(c')*.

Program for P_i:

01 **do forever**

02 **if** $pointer_i = null$ **and** $(\exists P_j \in N(i) \mid pointer_j = i)$ **then** $pointer_i = j$

03 **if** $pointer_i = null$ **and** $(\forall P_j \in N(i) \mid pointer_j \neq i)$ **and** $(\exists P_j \in N(i) \mid pointer_j = null)$ **then** $pointer_i = j$

04 **if** $pointer_i = j$ **and** $pointer_j = k$ **and** $k \neq i$ **then** $pointer_i = null$

05 **od**

Question 9 Socket API and TCP/IP (17 points):

9.a (2 p) Someone has suggested shortening the TIME_WAIT state duration. What could be the outcome of this suggestion?

9.b (1 p) How many simultaneous socket connections possible? What does it depends on?

9.c (3 p) Is congestion control in the Internet done through an end-to-end method or network-assisted method? Explain your answer.

9.d (3 p) Explain why this is so w.r.t question **9.c**.

9.e (5 p) Explain TCP's congestion control algorithm.

9.f (3p) What is the effect of TCP's congestion control and error control in real-time traffic? Explain your answer

---