Chalmers University of Technology
Computer Science and Engineering
Elad Michael Schiller

~~2013~~ 2014-01-17

Written exam in EDA387/DIT663 Computer Networks 2013-01-17. 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 (10 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 ns chalmers.se @dns.uu.se

```
; <<>> DiG 9.3.2 <<>> ns chalmers.se @dns.uu.se
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1628
;; flags: qr aa rd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL:7

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

;; ANSWER 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:
dns.uu.se.           14400   IN    A       130.238.7.10
dns.uu.se.           14400   IN    AAAA    2001:6b0:b:242::10
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: 44 msec
;; SERVER130.238.7.10#53(130.238.7.10)
;; WHEN: Thurs. Dec 10 20:19:56 2013
;; MSG SIZE  rcvd: 252
```

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

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

1.b (1p) To which DNS-server is the query sent? What is the IP address of this server?

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

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

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

1.f (1p) 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.

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

Question 2 IPv6 Addresses (6 points)

An IPv6 node has an Ethernet-interface with MAC address 5C-26-0A-66-77-7C.

2.a (3p) Rewrite each of the following IPv6 addresses using optimal zero-compression.

(i) 2001:06B0:0000:0000:5E26:0AFF:FE66:777C

(ii) FF02:0000:0000:0000:0000:0001:FF66:777C

(iii) FE80:0000:0000:0000:5E26:0AFF:FE66:777C

2.b (3p) What is the <u>type</u> and the <u>scope</u> of each one of the given addresses in question **(2.a)**.

Question 3 IPv6 operations (6 points)

3.a (3p) There is neither broadcasting nor ARP in the new version of the Internet Protocol IPv6. Describe the substituting <u>operation</u>, its <u>purpose</u>, the <u>protocol</u> and the <u>messages</u> used by an IPv6 node and its neighbors which are attached to an Ethernet LAN.

<u>**Hint:**</u> You are allowed to make use of the address (es) in the previous question if you would like to include example(s) in your answer.

3.b (3p) Describe clearly the stateless autoconfiguration of an IPv6 node that has an interface attached to an Ethernet LAN. In your answer mention in detail how the address will be configured, the protocol, the messages, and their relevant content, utilized in order to achieve this type of IPv6 configuration.

Question 4 (2 point)

6.a (1 p) An ____ E = (c(1), a(1), c(2), a(2), ...) , an ____ sequence, such that the ____ a(i) is ____ to ____ c(i-1) and results in ______ c(i), i.e., c(i-1) $\xrightarrow{a(i)}$ c(i) (i>1).

6.b (1 p) Please complete the definition of the mutual exclusion task, ME. (1) _____ one ______ can ______ it's ____ in any ____, and (2) every _____ can _____ it's _____ in _____ many ______ in every ______ in ME.

Question 5 (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.

```
upon a pulse
      forall Pj ∈ N(i) do send (j,clock_i)
      max := clock_i
      forall Pj ∈ N(i) do
                  receive(clock_j)
                  if clock_j > max then max := clock_j
      od
      clock_i := (max + 1) mod ((n + 1)d + 1)
```

4.a (2 p) The algorithm considers bounded set of values for the variable *clock*. This bound depends on the number of nodes in the network, *n*. Why that property is considered to be unattractive? Is the same true for the network diameter, d? In the context of the Internet, where is the difference?

4.b (4 p) Prove the correctness of the algorithm above.

Question 6 (4 points)

Below please find the spanning-tree construction algorithm. A variant of that algorithm is proposed. In this version every processor repeatedly checks whether the value of the *dist* variable of its parent in the tree is smaller than the value of its own *dist* variable. Processor pi does not execute lines 8 to 16 of the code when the above condition holds. Is the proposed algorithm a self-stabilizing spanning tree construction algorithm? Prove your answer.

```
Root: do forever
        for m := 1 to δ do write r_im := ⟨0,0⟩
    od
Other: do forever
        for m := 1 to δ do write lr_ml := read(r_mi)
        FirstFound := false
        dist := 1 + min{lr_mi.dis | 1 ≤ m ≤ δ }
        for m := 1 to δ
        do
            if not FirstFound and lr_mi.dis = dist -1
                write r_im := ⟨1,dist⟩
                FirstFound := true
            else
                write r_im := ⟨0,dist⟩
        od
    od
```

Question 7 (8 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 both Lemma 2.4 and Theorem 2.1, but there <u>*no need to prove*</u> *Lemmas 2.2 and 2.3*!

P$_1$: do forever

  if $x_1 = x_n$ then

    $x_1 := (x_1 + 1) \mod (n+1)$

P$_i$ $(i \neq 1)$: do forever

  if $x_i \neq x_{i-1}$ 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 p_i, x_i is not equal to j (Lemma 2.3)~~

(4 pt) 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)

(4 pt) 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 8 (6 points) `select()`

Each of the following parts of a program contains a flaw. Identify and describe the flaw in a few short sentences or points. You do not have to correct the flaw; you should just find and describe it! (Note: you're not looking for, e.g., syntax errors. Find conceptual flaws in the program.)

The following program has two sockets, `sockA` and `sockB`, from which the system expects to receive 24 byte messages. Each message is to be stored in a variable of type `message_t`, which is sufficiently large to contain the message. The messages are processed using the `process_message()` method. It is assumed that this method is capable of authenticating the message and checking that the message has the correct format. You can also assume that the `handle_*_error()` methods do something sensible.

```
/* includes, declarations, etc. */

static bool receive_message_from_socket( int sock ) {
      message_t msg;
      ssize_t receivedBytes = 0;
      // make sure that we receive the whole message
      while( receivedBytes < sizeof(msg) ) {
            ssize_t ret = recv( sock,
                  ((char*)&msg) + receivedBytes,
                  sizeof(msg) - receivedBytes,
                  0
            );
            if( 0 == ret ) {
                  // peer disconnected; close socket and return false to
                  // indicated that the connection has closed.
                  close( sock );
                  return false;
            }
```

```
            if( -1 == ret ) handle_recv_error();
            receivedBytes += ret;
        }
        // OK, process message and return success
        process_message( msg );
        return true;
}

int main() { /* initialization code has been omitted */
        while( sockA != -1 || sockB != -1 ) {
            // initialize read set
            fd_set readfds;
            FD_ZERO( &readfds );
            if( sockA != -1 ) FD_SET( sockA, &readfds );
            if( sockB != -1 ) FD_SET( sockB, &readfds );
            // call select(). parameters that are NULL (=0) are ignored by
            // select() - i.e. this is not an error!
            int maxfd = sockA;
            if( sockB > maxfd ) maxfd = sockB;
            int ret = select( maxfd+1, &readfds, 0, 0, 0 );
            if( -1 == ret ) handle_select_error();
            // check sockA for messages
            if( FD_ISSET( sockA, &readfds ) ) {
                bool stillOpen = receive_message_from_socket( sockA );
                if( !stillOpen )
                    sockA = -1;
            }

            // check sockB for messages
            if( FD_ISSET( sockB, &readfds ) ) {
                bool stillOpen = receive_message_from_socket( sockB );
                if( !stillOpen )
                    sockB = -1;
            }
        }
        /* de-initialization code has been omitted */
        return 0;
}
```

Question 9 TCP/IP (12 points):

9.a (3 p) What is the main idea of the TCP-friendly congestion control? Explain carefully (you do not need to write and explain the exact formula of the data-rate): outline the goal and how it is achieved.

9.b (3 p) Explain how can overlays be used in peer-to-peer file-sharing applications to address (i) searching and (ii) fetching of content. Accompany your explanations with examples.

9.c (2 p) Explain how can a connected dominating set be used in ad-hoc wireless networks in order to route messages between nodes.

9.d (4 p) Describe the marking algorithm for computing a connected dominating set and show its correctness.
