Chalmers University of Technology	(Dokumentnamn)	(Dokumentnummer)
Division of Networks and Systems	Written exam	
EMS	Vårt datum	Vår referens
	2012-08-22	
	Ert datum	Er referens

Written exam in EDA387/DIT663 Computer Networks 2012-08-22. Exam time: 4 hours.

Material allowed: **NONE**

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

Credits:	30-38	39-47	48-60
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 (3 points)

A cache-only local name server has received a standard query from a local client about type MX for the name "msn.com". Suppose that the server had no such information in the cache. Describe clearly the server steps while answering the query. Please use DNS terminology when describing the different server steps when trying to resolve the NS lookup.

Question 2 Ethernet and physical addresses

Assume that a host A has been inactive for a long time before a certain application generates IP datagrams to destination B with IP address 129.16.211.2. The host has Internet connection using an Ethernet NIC connected to the local network where each device is attached to one interface of an Ethernet switch. The host A has the following configuration.

```
C:\>ipconfig /all
Windows IP Configuration

Ethernet adapter Local Area Connection:

        Physical Address . . . . . . . . . : 00-1B-77-D3-20-B9
        DHCP Enabled . . . . . . . . . . . : Yes
        IP Address . . . . . . . . . . . . : 129.16.214.119
        Subnet Mask   . . . . .  . . . . . : 255.255.252.0
        Default Gateway . . . . . . . . .  : 129.16.212.23
        DHCP Server   . . . . . .  . . . . : 129.16.212.24
        DNS Servers . . . . . . . . . . .  : 129.16.1.53
```

a. **(1 point)** Are the source host A and destination B on the same subnet or not? Motivate and explain how you find the answer.

b. **(1 point)** Assume that there are no dynamic entries in ARP table at the host A. For what purpose does the host A need to get ARP mapping before transmitting the application packets?

c. **(2 point)** Describe the host's steps before and while sending the application packets. Give clear explanation of each step taken by the host in order to successfully transmit these packets.

d. **(1 point)** Explain how the entries in the MAC-address-table of the switch will be updated while host A is communicating in this case.

Question 6 (6 points) Self-stabilization

Show that in a synchronous uniform/anonymous ring, there is no deterministic self-stabilizing method for token circulating. Hint: The answer is similar to the leader election impossibility lemma, which is given below. For your connivance, we write the sentence number before each sentence. This way, you can write a shorter answer and just refer to the sentences that need to be changed.

Lemma (**Impossibility of Leader Election in Anonymous Networks**) **(s-1)** Let us consider an anonymous, synchronous network that has a topology of a fully connected graph. **(s-2)** No deterministic leader election method exists for this network.

Proof: **(s-3)** Let us assume, in a way of contradiction, that a deterministic leader election method, DLE, *does* exist and E is an execution of DLE that starts in configuration, $c[0]=<s_1,s_2,\ldots s_i, s_j,\ldots,s_n,r_1,r_2,\ldots r_i, r_j,\ldots r_n>$, where $s_i = s_j$ and $r_i = r_j$. **(s-4)** DLE solves the leader election task by reaching a configuration in which at least one processor has a state that is different than all other processors, which is the leader. **(s-5)** Therefore, in E there are two consecutive configurations $c[l_{last\text{-}same}]$ and $c[l_{first\text{-}different}]$ that are the last, and respectively, the first configurations in E for which $s_i = s_j$ and $r_i = r_j$ do, and respectively, do not hold, where $l_{first\text{-}different} = l_{last\text{-}same}+1$. **(s-6)** Therefore, DLE causes at least one processor to take a step from $c[l_{last\text{-}same}]$ to $c[l_{first\text{-}different}]$ that is different than the step of other processors between these two configurations. **(s-7)** This is a contradiction, because by definition any deterministic method implies that if $s_i = s_j$ and $r_i = r_j$ holds for all p_i and p_j in $c[l]=<s_1,s_2,\ldots s_i, s_j,\ldots,s_n,r_1,r_2,\ldots r_i, r_j,\ldots r_n>$, then both p_i and p_j take identical steps. □

Question 7 Question on `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.)

Hint: These programs uses `select()` and they are supposed to be non-blocking. Consider which operations can actually block the processes that execute these programs.

(7.a) (6 points) 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.

Question 3 Routing

a. **(2 point)** Describe clearly what "link state" is and what information is included when it is advertised by OSPF.

b. **(4 points)** What are the new features and enhancements in RIPv2 compared with RIPv1? Describe briefly.

Question 4 Congestion control

a. **(1 points)** Describe: What is network congestion?

b. **(2 points)** What is congestion control? How does congestion control differ from flow control?

c. **(2 points)** How is congestion controlled in the Internet? Briefly describe the method used. Which protocol takes care of it?

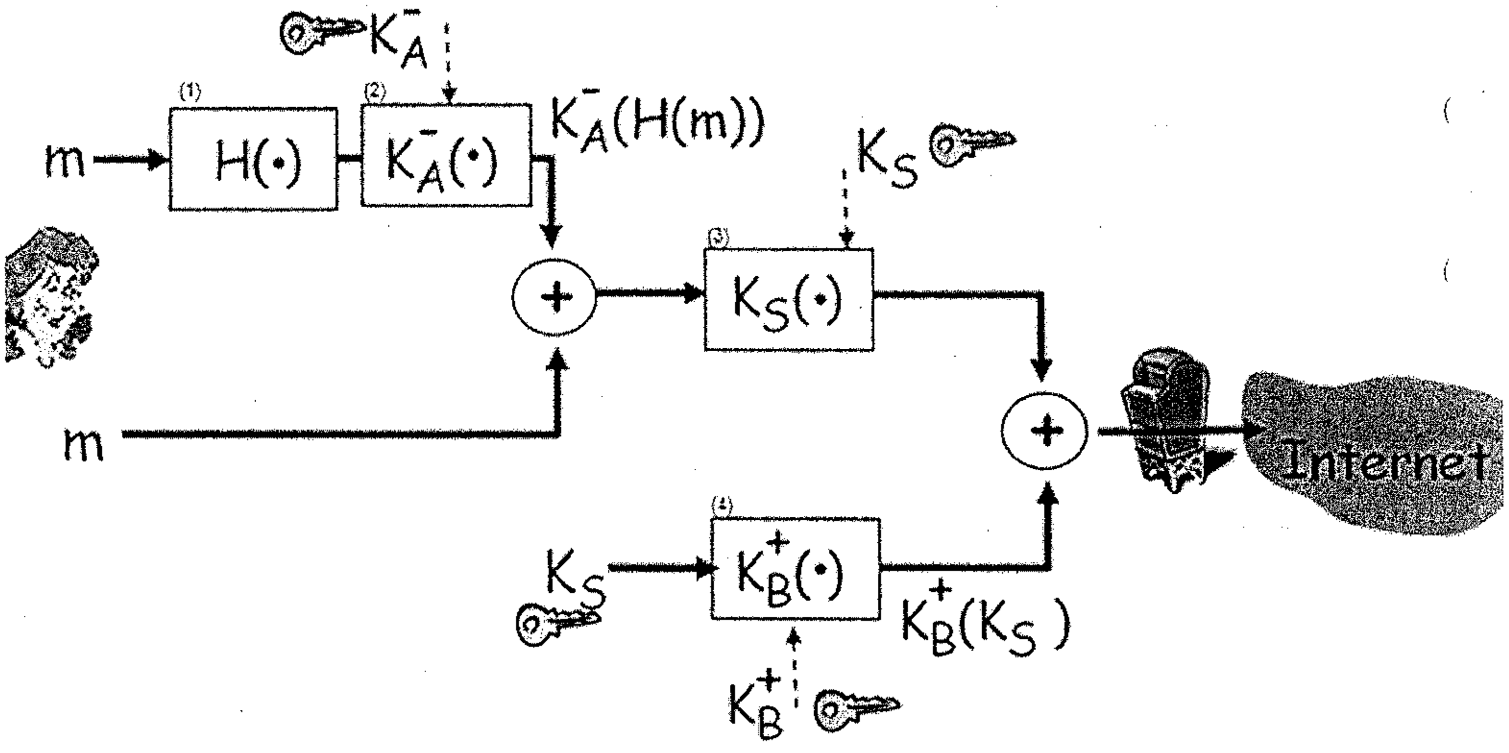
Question 5 Network Security

Please take a look at the figure in which Alice is preparing a message to Bob.

a. **(4 points)** For each rectangle from top left to bottom right, please explain the steps that Alice takes.

b. **(4 points)** Please draw the Bob's process when processing Alice's message.

c. **(4 points)** Please explain the steps that Bob must take when processing Alice's message (for each rectangle from top left to bottom right).



```
/* 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;
}
```

(7.b) **(6 points)** The following program accepts new connections using the `listenfd` socket. The first byte sent by a client is expected to be an 8 bit ID.

- You may assume that the `handle_*_error()` methods do something sensible.
- The helper method `register_client(client, id)` verifies the client ID is acceptable and if that is the case, enters the client into a global list. Otherwise it closes the connection.

- The method `add_client_sockets_to_readfds()` properly adds all active clients in the global list to the readfds. It returns the largest socket number it encounters.
- `handle_registered_clients()` handles clients that are ready to send data according to `readfds`, and removes clients that close their associated connections from the global list. No data is ever sent to the clients, the program only receives and processes data sent to it.

```
/* includes, declarations, etc. */
int main() {
        int listenfd = -1;
/* initialization code, such as setting up a listening socket on listenfd,
   has been omitted - this is not the error you're looking for */
        while( 1 ) {
                fd_set readfds; // initialize read set
                FD_ZERO( &readfds );
                int maxfd = add_client_sockets_to_readfds( &readfds );

                FD_SET( listenfd, &readfds );
                if( listenfd > maxfd ) maxfd = listenfd;

                int ret = select( maxfd+1, &readfds, 0, 0, 0 ); // call select

                if( -1 == ret ) handle_select_error();

                // is there a new client waiting?
                if( FD_ISSET( listenfd, &readfds ) ) {
                        sockaddr_in clientAddr;
                        socklen_t clientAddrLen = sizeof(clientAddr);

                        int client = accept( listenfd,
                                (sockaddr*)&clientAddr,
                                &clientAddrLen
                        );

                        if( -1 == client ) handle_accept_error();

                        // receive 8bit client ID
                        unsigned char id;
                        int ret = recv( client, &id, sizeof(id), 0 );

                        if( 0 == ret ) {
                                close( client );
                                continue;
                        }

                        if( -1 == ret ) handle_recv_error();

                        // register client
                        register_client( client, id );
                }

                handle_registered_clients(&readfds);//handle registered clients
        }
        return 0;
}
```

Question 8 Multiple Choice questions (11 points)

Instructions: Select the single correct choice (labeled X,Y,Z or W) among the available options. Each correct answer will give you 1 point. Zero points are given for a blank answer, or an incorrect answer.

Please write down the **question letter and the full text of the answer** so we can avoid confusion!

8.a. When an authoritative DNS server replies with answer it issues a TTL value for each RR.
[X] TTL value will be used to limit the number of hops across the DNS servers before answer reaches the client.
[Y] TTL value will be used by the cache to specify how long time the entry of RR may be reused before it is removed.
[Z] The authority's administrator specifies TTL value for each RR by which it limits life-time of the binding in the database.
[W] Cache-only and recursion DNS servers are allowed to choose their own TTL values for the entries they cache from answers.

8.b. How would the ICMP module in a host act if it receives an erroneous packet containing an ICMP error-message?
[X] Send back an ICMP type-12 message "Parameter Problem" reporting the error.
[Y] Send a request for retransmission of the same message.
[Z] Just ignore the problem.
[W] Send the packet back to the source.

8.c. Suppose that a router has two Ethernet interfaces connecting two LANs. A host A on LAN-1 wants to send an IP packet to host B on LAN-2, What will be done by host A before?
[X] Will search its ARP table for an entry for host B in order to map B's IP address to MAC-address.
[Y] Send an ARP request with the IP address of host B as target, so that host A will get an ARP-reply from the router's ARP-table.
[Z] Send an ARP request with the IP address of the router's interface on LAN-1 as target, so that host A will get an ARP-reply from the default gateway itself.
[W] Send an ARP request with the IP address of host B as target which will be forwarded by the router toward LAN-2 so that host A will get back an ARP-reply from host B through the router.

8.d. If an ARP request arrives, the software in receiving host extracts the sender's address pair.
[X] If the receiving host is the target, it will add the sender's address pair to its ARP-table if the mapping does not already exist.
[Y] If the receiving host is not the target, it will refresh the sender's address pair in its ARP-table if the mapping does not already exist.
[Z] If the receiving host is not the target, it will add the sender's address pair to its ARP-table if the mapping does not already exist.
[W] All hosts are allowed to add as well as refresh their entries in ARP-table when any ARP reply arrives regardless of the destination.

8.e. A CIDR block of addresses consisting of four class C contiguous network addresses will have subnet mask:
[X] 255.255.255.0
[Y] 255.255.254.0
[Z] 255.255.253.0
[W] 255.255.252.0

8.f. An Ethernet switch is able to forward frames to their destinations by:
[X] Using ARP-table that contains mapping of Internet addresses to MAC addresses.
[Y] Querying all connected hosts to send their addresses to the switch.
[Z] Using tables of MAC addresses that are learned dynamically.
[W] Using configuration files containing forwarding tables.

8.g. The special IPv4 address `INADDR_ANY` can be used in what context in relation to TCP/IP communications?

[X] An outgoing connection can be made to `INADDR_ANY` in order to connect to any host.

[Y] A listening socket can be bound to `INADDR_ANY` to accept incoming connections from any machine with an IPv4 address.

[Z] A listening socket can be bound to `INADDR_ANY` to accept incoming connections on any interface on the local machine.
[W] INADDR_ANY is specified as the remote peer's address by `accept()` if `accept()` is unable to determine the real address of the remote peer (because, for example, the remote peer prefers to be `AnoNYmous`).

8.h. Which set of calls is used to establish an outgoing TCP connection, for example, from a TCP/IP client?

[X] `socket(), bind(), listen()`

[Y] `connect(), accept()`
[Z] `getaddrinfo(), bind(), socket()`
[W] `socket(), connect()`

8.i. Over a stream oriented connection, such as a TCP connection, a peer sends a 120-bytes (octets) long data stream. The receiving side attempts to `recv()` 64 bytes:

```
    ssize_t ret = recv( sock, bufferPointer, 64, 0 );
```

Which of the following is a possible outcome of this operation?
[X] The recipient receives 64 bytes (ret = 64), and 56 bytes are discarded
[Y] The recipient receives 60 bytes (ret = 60), and 60 bytes can be received with another call to `recv()` at a later time.
[Z] The recipient receives 68 bytes (ret = 68), and 52 bytes can be received with another call `recv()` at a later time.
[W] The recipient receives 64 bytes (ret = 64), but the entire 120 byte long data stream is kept and can be received with `recv()` as a whole at a later time.

8.j. In relation to blocking and non-blocking modes set on sockets used in TCP/IP communications, which of the following statements is *not* true?
[X] In non-blocking mode, `recv()` will return an error if no data is available to be received.
[Y] In non-blocking mode, `connect()` may return an error condition (i.e., it may return -1), but the connection can still succeed at a later time.
[Z] In blocking mode, `recv()` may receive and return less data than what was requested of it.
[W] In blocking mode, `send()` will wait (i.e., block) until the remote peer has successfully received all data that is to be sent to it.

8.k. The `select()` method can be used to detect events affecting one or more sockets. Sockets can be added to three sets, depending on what events are to be detected. One of the sets is the `readfds`.

```
            int select(int nfds, fd_set *readfds, fd_set *writefds,
                        fd_set *exceptfds, struct timeval *timeout);
```

Which of the following is an event that is never detectable using `select()` and the `readfds`?
[X] The connection associated with the socket has closed.
[Y] A non-blocking attempt to establish a connection has completed (but the peer has not yet sent any data and/or closed the connection).
[Z] A new connection is ready to be accepted from the socket.
[W] Data is ready to be received from the socket.