CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
EDA-387	COMPUTER NETWORKS			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
EDA-387-7		27/8/2015	13	4

Solved task Behandlade uppgifter No/nr		Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylles av lärare.
1	1	6	
2	2	5	
3	3	7	
4	4	5	
5	5	2	
6	C	7	
7	7	5	
8	8	2	
9	9	½	
10	10	2.5	
11			
12			
13			
14			
15			
16			
17			
18			
Total examination points Summa poäng på tentamen		42	37

3

1 a) The user wanted to ask about the authoritative server (ns1.ntnu.no) about the ~~mail~~ hostname of the mail exchanger that serves the 'ntnu.no' domain.

b) The first part of the question ntnu.no is the domain name <the user is querying.>

MX (is Mail Exchanger) the type of resource record we seek to find.

c) hostname: ns1.ntnu.no

IP-address: 129.241.0.208

Yes it is an authoritative server because it is used in the authority section. Also the flag has aa implying it is an authoritative answer.

d) Yes there is an answer in the reply and status is NOERROR.

The Reply gives the ~~mail~~ hostname (mx0.ntnu.no) of the mail exchanger responsible for the ntnu.no domain with a preference value of 10.

The type of answer is MX (mail exchanger)

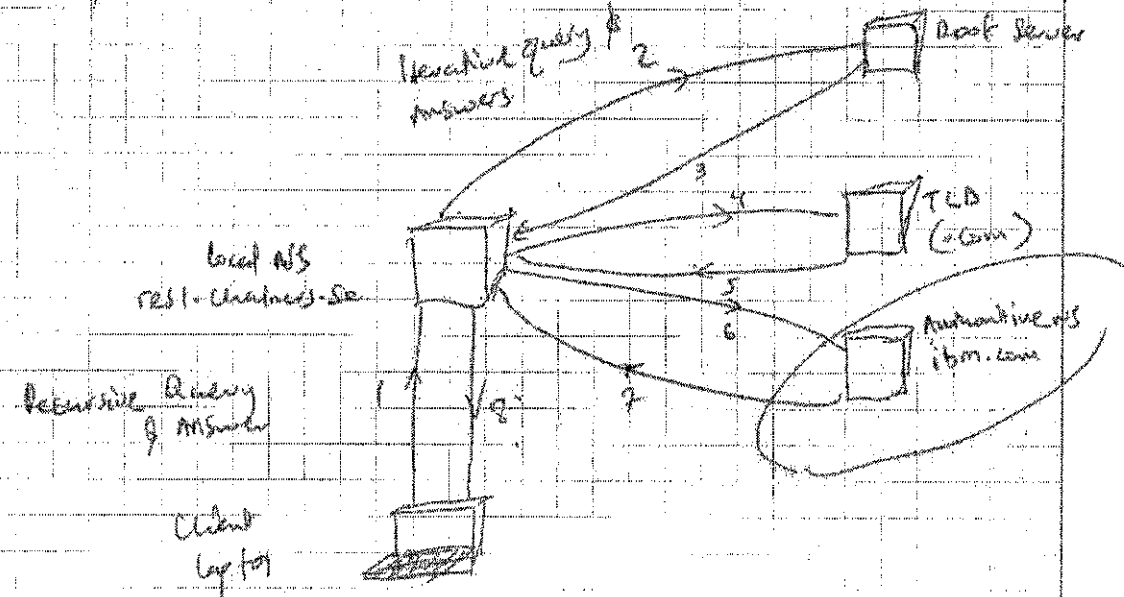
The Total Time To Live TTL = 60 seconds the time for which the answer is valid before it can be refreshed.

2

3

2a) The client has queried to get information about hostname of the nameserver that can give information about the domain that hosts ibm.com

2b)

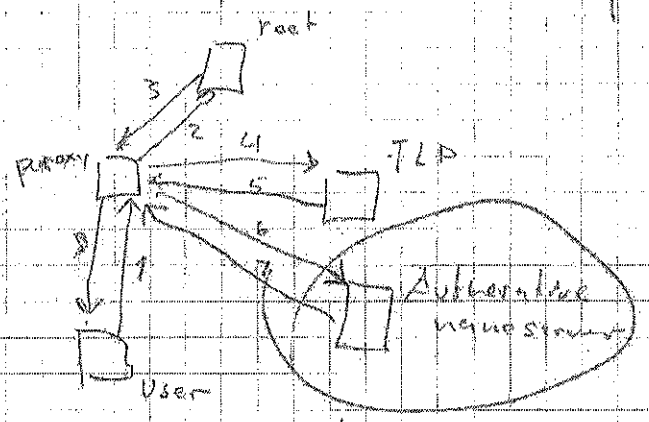


1. Client makes recursive query to local NS (res1.chalmers.se)
2. Since local NS does not have the answer, it makes iterative DNS queries as follows
   2. local NS contacts root nameserver about top level domain (TLD) Server of .com
   3. Root NS answers with the TLD (.com) domain server
   4. local NS queries .com about server ibm.com
   5. .com Server responds with answer for the ibm.com authoritative NS. enough here)
   6. local NS queries authoritative NS (ibm.com)
   7. ibm.com responds with the Ip adress of the server and
      - stores the record in its cache valid for a given TTL. 3
   8. local NS gives ip address (recursive answer) to the client.

2c) After obtaining the DNS information, the local server stores it in its cache and the answer is valid for a given TTL value. 1

3

2 a, The client wants to know the NS=name server of ibm.com domain



b,

- If the name server is not cached in the proxy the proxy will ask the root server for <which TLD (Top level domain) to ask for ".com"> ?!
- Then when the TLD is known, the proxy will ask it for which Authoritative nameServer to talk to.
- The Authoritative nameserver replies the full name server, about itself!!
- Proxy, replies an answer to the query made by the user. 2

This is done with UDP

The querys and replies are done recursive and iterative. when?

c, The local server will store the NS in its cache, so that it can be used next time to save time. TTL

3a) i) 2001:0660:0002:0410:0000:0000:0000:0001

ii) FF02:0000:0000:0000:0000:0001:ff6C:14dd

iii) FE80:0000:0000:0000:020c:f1ff:fe6c:14dd 1

b) 2001:660:2:10::1 is type unicast

FF02::1:ff6C:14dd is type multicast

FE80::20c:f1ff:fe6c:14dd is type unicast

Unicast implies its an IPV6 address for a given interface.

multicast implies it is an IPV6 address to a ~~grou~~ a given group of interfaces. delivering packeet to - - 2+

c) FF02::1:ff6C:14dd can not be used as a valid source address in IPV6 packet because it is a multicast address that handles a group of interfaces and hence can only be used as destination address. 1

d) 2001:660:2:10::1 has scope global

FF02::1:ff6C:14dd has scope ~~link~~ local

FE80::20c:f1ff:fe6c:14dd has scope link local

Scope global implies that it can be used to route packets over to the internet.

link local implies that packets ~~are~~ can only be routed within a given network connected by physical interface.

packets from these addresses can not be routed across a router. 2+

3

e) IPv6 Neighbour discovery is used to determine the link local address of a neighbour's interface. (MAC) IPv6 layer 5

It uses ICMPv6 messages and ~~solicited~~ solicited-node multicast addresses encapsulated in IPv6 packets to determine the link local address of a neighbour. 6

The ICMP messages used are neighbour solicitation message and neighbour advertisement message. operation?

f) IPv6 Neighbour discovery uses neighbour solicitation message and neighbour advertisement message.

- For Neighbour solicitation message:

In a layer-2 PDU we have the encapsulated frame with

Source: link layer local address of A (unicast)

destination: multicast address of B

In a layer-3 PDU we have the encapsulated packet with

Source: IPV6 address of A

destination: IPV6 multicast address of B eg FE80::20C:1FFF:FE6C:14DD ? eg. FF02::1:FF6C:14DD. ?

- For Neighbour advertisement message Same ?!

In a layer 2 - PDU we have the encapsulated frame with

Source: link local address of B (unicast)

Destination: link local address of A (unicast)

In a layer 3 - PDU we have the encapsulated frame with

Source: IPV6 unicast address of B

Destination: IPV6 unicast address of A. 1

3

In the function if(FD_ISSET(listenfd, &readfds)) { ?

There are 2 system calls that cause blocking

- accept()
- recv().

A call to accept causes blocking in the function until either an error has occured or a connection file descriptor, client has returned successfully.
This blocks all other clients trying to make connection with the server.

A call to ~~receive~~ recv() can also cause blocking. less than 8bits

If there is no data in the buffer ~~to~~ to be read and the ~~non-blocking~~ non blocking flag is ~~not~~ not set, the call will block until data is available to be read.

If there is no data in the buffer to be read and the non-blocking flag is ~~not~~ set, the call will return an error and the ERRNO will be set accordingly.

3.

a) Adjustment: p_i does not ① adjust its clock.

Agreement: p_i's clock ② agrees with the clock estimate of ③ all other correct processors that have also ④ been working correctly for atleast K time units. 2p.

b) At the starting configuration, P will read every counter and check of its neighbours.

Suppose P has been executing correctly and continues to execute correctly for successive steps of K; the set NB will not be empty because atleast P will be in the set and continue to be in the set as long as it executes more than K=2 success steps correctly.

This is because it is not behind any other processor.

P continues to update all neighbourg processors of the order given by Q.counter[P] = Q.counter[P] + 1 (mod 3).

No, the question was only $R = NB \neq \emptyset$ after the first step. 0p

3

6 a) d is the diameter of the network.

(2) n is the number of nodes (clocks) in the network.

b) converge-to-the-max depends on 'd' and 'n' whereas converge-to-the-min does not depend on 'n'

In an actual system, where the number of nodes, n, increases at a much higher rate than the diameter hence converge-to-the-max gets subject to more values, M, for the bound which is not good.

(1)

converge-to-the-mean is better for scalability because it is not affected by increase in node number of nodes, n.

c) converge-to-the-max is easier to work with at application layer.

(1) This is because converge-to-the-min converges to a past clock value which can confuse the application.

d) 1. wraps around the zero value and no processor assigns zero to its clock value

2. d ✓

3. induction proof ✓

4. wraps around the zero value and assigns zero to its clock value

5. d ✓

6. M-d ✗

7. d ✓

8. case holds ✓

(3)

3

a) We prove this by way of contradiction. Suppose that in the starting configuration C_1 P_1 does not change the value of x_1.

- Let C_2 be the configuration that immediately follows C_1 in which P_2 is the only processor that can change the value of its variable x_2, leading to $x_1 = x_2$ and $x_2 \neq x_n$ remains for the remaining $n-2$ rounds.

- Let C_3 be the configuration that follows immediately after C_2 such that only P_3 can change its register value $x_1 = x_2 = x_3$.

In general, the only processor that can change the value of its register for a given configuration C_{i-1} is P_i such that $x_1 = x_i$ but $x_i \neq x_n$ for the next $n-i$ rounds.

In the n^{th} round, $x_n = x_1$ and therefore P_1 changes the value of x_1 once in n rounds. contradiction □.

b)

3

(7 b) Let C be the starting configuration for every fair execution.

From Lemma 2.2, C is a safe configuration from which all X variables are equal.

Let C_1 be the configuration that follows C.

- Using Lemma 2.4, Processor P_i changes the values of its variable at least once in n rounds.
- Using Lemma 2.3 and knowing that in any round, only one processor can change the value of its register. The complexity of all $P_i (i \neq 1)$ changing their values is one in n rounds

Therefore within $O(n^2)$ rounds all the processors including P_1 have changed the value of their registers in relation to ME □

3

Lemma: Suppose that in an anonymous, synchronous uniform network of ring topology, there is no deterministic self-stabilising algorithm for token circulation, DTC. The processors take a step at the same time, each starting from the first line of the algorithm and controlled by a central deamon.

Proof

We prove this by way of contradiction that suppose that a deterministic self-stabilising algorithm DTC for token ring circulation exists.

Let the processors start in a configuration C_0 such that the ~~starting~~ starting execution is given by state S_i and register value r_i

$E = \langle C_0, S_1, S_2 S_3 \cdots S_n\ r_1\ r_2\ r_3 \cdots r_n, C_1, C_2 \rangle$ where by $S_i = S_j$

and register values $r_i = r_j$ for ~~four~~ successive configurations and $i \neq j$.

The task of the DTC is to reach a configuration where all values of the registers are the same such that $r_i = r_j$ and $S_i = S_j$

Let $C_{\text{last-same}}$ and $C_{\text{first-different}}$ be the consecutive configurations for which $r_i = r_j$ and $S_i = S_j$ respectively hold and do not hold such that $C_{\text{first-different}} = C_{\text{last-same}} + 1$.

It is clear that the DTC leads to a ~~configuration~~ set execution set in which two respective registers ~~not~~ hold different values.

3

For a deterministic self-stabilizing algorithm for token ring orientation, we should have $s_i = s_j$ and $r_i = r_j$ since the processors are identical and they take the same steps in execution.

Therefore the DTC causes atleast one of the processors to execute a different step than all other processors and also have a different register value compared to all other processors which is a contradiction □

3

a) A set of legal executions for a given algorithm is a given set of configurations run in fair execution and all ~~belong~~ exhibit a legal behavior.

Safe configuration for a given algorithm is one that is reached after a set of fair executions all of which belong to the set LE (Legal Execution)

b) No, the algorithm is not self-stabilizing

Stopping configuration c when

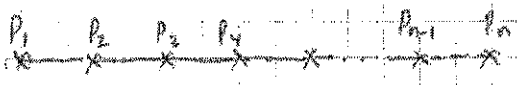
leader $\doteq$ (id = minimum{id_m | $1 \le m \le n$})

3

a) The task of vertex color is to give colors to the nodes of a graph such that no two neighbouring nodes P_i and P_j $|j \neq i$ have the same colour.
There is no upper bound on the number of colours that can be used.
The task can use $(\Delta+1)$ colours where Δ is the upperbound on the number of colours.

b) The algorithm converges at the $O(n)$ where n is the number of nodes.

Consider a set of nodes connected to each other in form of a line as show below.



Suppose that the link between P_{n-1} and P_n is not operative.

If the the link P_{n-1} to P_n recovers and assigns a $color_n$ such that $color_x == color_{n-1}$, then every node in the system, P_i $|$ $1 \leq i \leq n$ will run the algorithm to change color hence $O(n)$.