Chalmers University of Technology

Computer Science and Engineering
Elad Michael Schiller

2021/10/28

Written exam in EDA387/DIT670 Computer Networks 2021-08-26. Exam time: 4 hours.

Examiner: Elad Michael Schiller, phone: 073-6439754

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-speaking 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.
10. Do not hand in anything that is not part of the solution.
11. Always write your answers in your own words, using your own diagram, etc. In case you have to use the words of others, e.g., a formal definition, please make sure to use quotation marks and clearly identify the source.
12. Absolutely do not collaborate in any way or form with other people during the entire exam time (regardless of whether you or they are still writing the exam or not).

Question 1: General Questions (28 points in total)

For each of the following statements, please state true or false. In case you are not sure about your answer or the statement is not clear to you, you are welcome to justify your claim briefly (but you do not have to). In other words, a *correct* simple "True" or "false" answer receives 2 points per item. An incorrect answer without an explanation receives 0 points. An incorrect answer with an explanation receives between 0 and 2 points.

(1.a) **(2 points)** The end-to-end design principle played a key role in the success of the Internet since it encouraged the rapid and extensive adaptation of state-of-the-art routing technologies. False

(1.b) **(2 points)** End-to-end protocols and link-layer protocols solve different problems using completely different algorithms. False

(1.c) **(2 points)** Consider a common implementation of TCP/IP. Suppose the receiver sends an acknowledgment for packet x to the sender and that acknowledgment is lost. TCP has mechanisms that guarantee that the Sender always retransmits packet x. False

(1.d) **(2 points)** Due to the invention of the congestion control mechanisms, the throughput of the Internet increased by 5 orders of magnitudes. This is because the Sender can automatically slow down its transmission rate in order to avoid overflowing (flooding) the Receiver's buffer. False

(1.e) **(2 points)** The Receiver can safely detect congestion whenever it receives more packets than the number of packets that its buffer can hold. False

(1.f) **(2 points)** Whenever the Receiver detects congestion, it singles to the Sender by transmitting duplicated ACKs. False

(1.g) **(2 points)** Routing Information Protocol (RIP) was the first routing protocol for the Internet (started in 1969 and used extensively until the end of the 1980s). However, due to RIP's communication costs, other protocols, such as OSPF and IS-IS are currently used more often. False

(1.h) **(2 points)** A recursive DNS query requires the client to wait (i.e., the calling process needs to be blocked by the kernel) until the answer arrives (e.g., with the server's IP address) whereas an iterative query does not impose such a delay. False

(1.i) **(2 points)** The design of the DNS is based on the fact that the dataset (that includes hostnames and their IPs) is read-mostly and that it has weak consistency guarantees. True

(1.j) **(2 points)** In class, we showed that there is no uniform digital clock-synchronization algorithm that uses only a constant number of states per processor. The Network Time Protocol (NTP) uses a constant number of 64-bits to represent the clock values. Such contradictions can exist between theory and practice. False

(1.k) **(2 points)** The fact the problem X does not have a self-stabilizing solution implies that it does not have a super-self-stabilizing solution. True

(1.l) **(2 points)** The success of software-defined networks (SDNs) happened because it made the management of layer 2 protocols much more cost- efficient. True

(1.m) **(2 points)** Consider the socket API, i.e., socket(), bind(), listen(), accept(), connect(), accept(), select(), close(), and shutdown(). Using this API, it is possible to implement a lock (mutex). True

(1.n) **(2 points)** Consider the socket API, it is possible to implement a timer. True

.....

Question 2: self-stabilizing distance-2 coloring using $O(\Delta^2 \log N)$ bits registers (20 points in total)

Let $P=\{p_1,p_2,\ldots, p_n\}$ be a set of nodes in a network whose topology is of a general graph, $G:=(P, E)$, where $E=PxP$ are edges that represent the communication links. Assume that $\Delta \geq \max \{\delta_i : p_i \in P\}$ is an upper bound on the degree, δ_i, of every node p_i. In the task of distance-2 coloring on G, the aim is to assign a color x_i to each node $p_i \in P$, such that any two nodes p_i, $p_j \in P$ at distance at most two in G are assigned different colors $x_i \neq x_j$, where each color is selected by the algorithm from a finite set, $C \subseteq \mathbb{Z}^+$, of color palette.

2.a (4 points) Suppose that the network is anonymous, i.e., all nodes run the same algorithm, and the nodes are not assigned with globally unique identifiers. Please show that there is no deterministic solution to the task of distance-2 coloring on G. Consider the case of two node and show that starting from a state in which all nodes have the same colors does not help to break symmetry.

Recall the definitions of fair schedulers such as a central and a distributed demon. A central demon ensures that no more than one processor is enabled to execute at each instant of time. A distributed demon enables any set of processors at any instance of time. If you decide to assume central demon for question items 3.c and 3.d, then your *correct* answer can receive 5 points per item. If you decide to assume distributed demon for question items 3.c and 3.d, then your correct answer can receive at most 8 points per item.

2.b (5 to 8 points) Suppose that the color palette C includes exactly Δ^2+1 different colors and that each node in P is assigned with a globally unique identifier in $\{1,\ldots,N\}$ that are represented by $\log_2 N$ bits, where $N \geq n$ is a bound on the number of nodes in P. Also, suppose that each node, $p_i \in P$, can write to a single register, from which all neighbors can read. Each register has the size of $O(\Delta^2 \log N)$ bits. Please propose a self-stabilizing algorithm for solving the task of distance-2 coloring on G. Please provide clear and exact pseudocode for the entire algorithm that you propose. State all of your assumptions clearly, and in case, you use solutions (sub-solutions) learned during the course, please name them clearly. (You also have to explain how to combine these sub-solutions with your own contribution.) There is no need to provide code that is not originally yours.

Every node write both his id and colors as well as the colors and ids of all of its direct neighbors. Then, the ideas can be derived from the distance one coloring solution.

2.c (5 to 8 points) Please prove the correctness of your proposed solution. That is, show that when starting from any system state (configuration), the algorithm constructs a distance-2 coloring on G. Please clearly state your assumptions.

The proof (of the distributed demon) in by induction of the ids --- similar to the proof of the distance-one solution.

.....

Question 3: self-stabilizing β-maximal-partitioning using *O*(log N) bits registers (22 points in total)

Let $P=\{p_1,p_2,\ldots, p_n\}$ be a set of nodes in a network whose topology is of a general graph, $G:=(P, E)$, where $E \subseteq P \times P$ are edges that represent the communication links. Let $\beta \in \{1,\ldots,N\}$ be a positive integer, where $N \geq n$ is an upper bound on the number of nodes degree. Suppose that each node in P has a globally unique identifier in $\{1,\ldots,N\}$ that can be represented using $\log_2 N$ bits. Let $P_1,P_2,\ldots, P_k$ be a partitioning of P, such that $\forall_{x, y : x \neq y} P_x \cap P_y = \emptyset$, $\bigcup_x P_x = P$, and $\forall_x P_x \neq \emptyset$ as well as $G_x:=(P_x, E_x)$ is a (connected) tree rooted at p_x and the distance in G between p_x and any other node in P_x is at most β, where $\cup$ is the set union operator, $|A|$ is the cardinality operator applied to the set A, $x, y \in \{1, \ldots, k\}$ and $E_x \subseteq E \cap (P_x \times P_x)$. In this case, we say that $P_1,P_2,\ldots, P_k$ is an β-partitioning of G --- the order in which the partition sets are written does not play a role. Suppose that there are no $x, y \in \{1,\ldots,k\} : x \neq y$, such that $P_1, P_2,\ldots, (P_x \cup P_y), \ldots, P_k$ is also an β-partitioning of G. In this case, we say that $P_1,P_2,\ldots, P_k$ is an β-maximal-partitioning of G. Your task is to design a self-stabilizing algorithm for constructing an β-maximal-partitioning of G.

3.a (3 points) For the following questions, please provide clear arguments to support your claims.
3.a.i (1 pt) Please suggest at least one application to the problem of β-maximal-partitioning of *G*.
3.a.ii (1 pt) In what way can the parameter β play a role in your application?
3.a.iii (1 pt) In what way does the maximality requirement plays a role?

3.b (1 point) Please define the set of legal executions for the given task.

Every configuration solves the task

Suppose that each node, $p_i \in P$, can write to a single register, from which all neighbors can read. If you decide to assume that n=N and/or that each register has the size of *O*(N log N) bits when answering question items 4.c and 4.d, then your *correct* answer can receive 5 points per item. If you decide that N is much bigger than n and that each register has the size of *O*(log N) bits when answering question items 3.c and 3.d, then your correct answer can receive at most 9 points per item.

3.c (5 to 9 points) Please provide clear and exact pseudocode for the entire protocol that you propose. State all of your assumptions clearly, and in case, you use solutions (sub-solutions) learned during the course, please name them clearly. (You also have to explain how to combine these sub-solutions with your own contribution.) There is no need to provide code that is not originally yours.

Use the algorithm for leader election in a general graph. Do not let the flouting distances to grow up to N --- stop them at β.

3.d (5 to 9 points) Please prove the correctness of your proposed solution. That is, show that when starting from any system state (configuration), the algorithm constructs an β-maximal-partitioning of *G*. Please clearly state your assumptions and every step of the proof.

Similar to the proof of the original algorithm.