Informatique théorique et applications

Krzysztof Diks
Adam Malinowski
Andrzej Pelc

Token transfer in a faulty network

Informatique théorique et applications, tome 29, nº 5 (1995), p. 383-400
<http://www.numdam.org/item?id=ITA_1995__29_5_383_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 29, n° 5, 1995, p. 383 à 400)

TOKEN TRANSFER IN A FAULTY NETWORK (*)

by Krzysztof DIKS (1) (**), Adam MALINOWSKI (1) (***)
and Andrzej PELC (2) (****)

Communicated by W. RITTER

Abstract. – *A token originally situated in a given fault-free node of the complete network, called the source, has to visit all other fault-free nodes. Links and/or nodes of the network fail independently with probabilities* $p < 1$ *and* $q < 1$, *respectively. In a unit of time every node can be involved in at most one transmission; transmissions along a faulty link or involving a faulty node do not succeed. We consider various communication models depending on the ability of nodes to modify their behavior according to the outcome of previous transmissions. For all models we present token transfer algorithms working fast and with probability of correctness exceeding* $1 - n^{-\varepsilon}$, *where* n *is the number of nodes and* ε *an arbitrary positive constant.*

Résumé. – *Un jeton, situé d'abord dans un nœud fonctionnel d'un réseau complet, appelé la source, doit visiter tous les autres nœuds fonctionnels. Les liens et/ou les nœuds du réseau tombent en panne avec probabilités* $p < 1$ *et* $q < 1$ *respectivement ; toutes les pannes sont indépendantes. Pendant une unité de temps chaque nœud peut participer à au plus une transmission ; les transmissions dans lesquelles participe un nœud ou un lien défectueux n'ont aucun effet. Nous considérons plusieurs modèles de communication selon la capacité des nœuds à modifier leur comportement compte tenu des résultats des transmissions antérieures. Pour tous les modèles nous présentons des algorithmes de transfert du jeton qui sont à la fois rapides et qui travaillent correctement avec probabilité plus grande que* $1 - n^{-\varepsilon}$, *où* n *est le nombre des nœuds et* ε *est une constante positive quelconque.*

1. INTRODUCTION

Token transfer can be considered as a variation of the well-known broadcasting problem. In broadcasting, one node, called the source, has to transmit a message to all other nodes. [14] is an extensive survey of

(*) Received June 7, 1993; revised December 8, 1993; accepted February 3, 1994.

(1) Instytut Informatyki, Universytet Warszawski, Banacha 2, 02-097 Warszawa, Poland.

(2) Département d'Informatique, Université du Québec à Hull, C.P. 1250, succ. "B", Hull, Québec J8X 3X7, Canada.

(**) Partly supported by grant KBN 2-2043-92-03. This work was done during the author's stay at the Université du Québec à Hull supported by NSERC International Fellowship.

(***) Partly supported by KBN 2-1190-91-01.

(****) Partly supported by NSERC grant OGP 0008136.

the domain of broadcasting and closely related gossiping. Broadcasting and gossiping in networks with faulty links and/or nodes have been recently studied by many authors [1, 2, 4-6, 9-12, 15]. While the classical approach assumes an upper bound on the total number of faults and their worst case location [1, 11, 12], probabilistic models, where links and/or nodes fail independently with constant probability, have recently gained growing attention [2, 4-6, 9, 10, 15]. The goal in this case is the design of efficient algorithms which work correctly with high probability.

The special characteristic of token transfer, which distinguishes it from classical broadcasting, is that the same token has to visit all (fault-free) nodes in a sequential way, while in broadcasting all nodes that are already informed can disseminate the source message in parallel. Thus token transfer, even in fault-free networks, requires linear time, unlike classical broadcasting which can be done in logarithmic time. A similar variation of broadcasting has been considered in [3, 8] under the name of linear broadcasting: the task considered there was visiting every node by one of several tokens originally stored in the source. Thus our token transfer problem can be viewed as linear broadcasting with a single token.

We work under two alternative fault assumptions: in one of them nodes are fault-free and links fail independently with constant probability $p < 1$; in the second, links and nodes of the network fail independently with probabilities $p < 1$ and $q < 1$, respectively. The scenario with faulty nodes and fault-free links is not discussed, since in this case an asymptotically optimal algorithm always working correctly is trivial. In both cases faults are assumed permanent and of crash type: a transmission involving a faulty link or node does not succeed.

Under each of those fault scenarios we consider four communication models based on different ability of nodes to adapt their behavior according to success or failure of previous transmissions. Models range from non-adaptive, where communication scheduling is completely rigid, to adaptive which are characterized by large flexibility of transmissions. These models are precisely defined in section 2.

As usual in the theory of broadcasting and gossiping, we assume that in each time unit a node can be involved in at most one transmission. Our goal is the design of fast token transfer algorithms which have the following reliability property: given a positive constant ε, the algorithm works correctly in n-node networks with probability exceeding $1 - n^{-\varepsilon}$. Such algorithms are called ε-safe. For each model and every $\varepsilon > 0$ we present a fast ε-safe algorithm: time complexities of our algorithms range from $O(n)$ to $O(n \log n)$.

The paper is organized as follows. In section 2 we give a precise description of our communication models and some preliminary probabilistic facts used later on. Section 3 is devoted to the fault-free node scenario, while in section 4 we study the assumption of both link and node failures. Section 5 contains conclusions and open problems.

2. MODEL DESCRIPTION AND PRELIMINARIES

The communication network is represented as a complete n-node directed graph whose vertices are nodes of the network and arcs unidirectional communication links. Nodes are labeled with integers $1, \ldots, n$ and the arc from v to w has label vw. The node with label 1 is called the source. All nodes know all labels.

Our algorithms are synchronous: processors use a global clock. In one time unit every node can be involved in at most one transmission: it can call or be called by at most one other node, these two possibilities being exclusive.

We consider two fault scenarios. In the first, nodes are fault-free and links fail independently with probability $p < 1$. In the second, links and nodes other than the source fail independently with probabilities $p < 1$ and $q < 1$, respectively. It should be stressed that failures of links vw and wv are also independent. The source is assumed fault-free. All faults are permanent (the fault status of a component does not change during algorithm execution) and of crash type: faulty nodes do not attempt transmissions and faulty links do not transmit. We assume that a fault-free node which made an unsuccessful transmission attempt knows that the transmission failed but does not known a priori if this was due to a faulty link or faulty destination node. Likewise, a fault-free node which expected transmission from a node v at a given time unit and did not get it, does not know whether the node v or the respective link failed.

For each of the above fault scenarios, we consider four communication models based on different degree of adaptivity of communication. By this we mean the ability of nodes to modify their behavior according to the outcome of previous transmissions. The most rigid is the non-adaptive model NA: all transmissions must be scheduled in advance and whenever a node v scheduled to transmit to w has the token, it must send it. (This attempt may be unsuccessful due to a link or destination node failure and in this case the token does not move). In the model NA there is only one elementary transmission procedure SEND (v, w) which consists in an attempt of sending the token from node v to node w. The token moves to w if at the time of execution of SEND (v, w) it is in node v and if nodes v, w and the link vw are fault-free.

The next model is semi-adaptive (SA). Here transmissions are scheduled in advance as before, but nodes have the ability of attempting transmissions without trying to send the token, even when they have it. Such "idle" transmissions will prove useful for testing which nodes and links are fault-free, without risking the loss of control over the token. In the SA model two elementary transmission procedures are used: SEND (v, w) has the same meaning as before and CALL (v, w) consists in the attempt by v to call w without trying to send the token, even if the token is currently in node v. Clearly no move of the token results from this procedure: the only advantage is the increase of knowledge about fault status of other nodes and links. It should be stressed that a successful CALL (v, w) procedure does not involve sending any information (apart from implicit information that it was successful).

Finally, the most flexible models are adaptive ones: nodes can freely decide to which nodes they should attempt transmissions and whether a particular transmission should be of SEND or of CALL type. We distinguish two adaptive models: the restricted adaptive model RA and the general adaptive model GA. In RA only the node currently holding the token can attempt transmissions (of SEND or of CALL type), while in GA all nodes can attempt transmissions at all times (provided that every node is involved in at most one transmission at a time).

It should be noted that in models SA, RA and GA all decisions of nodes have to be based on the local history of the node (the success or failure of previous transmissions involving that node): we do not assume the existence of any central monitor supervising the execution of algorithms.

Combining these four communication models with the link failure scenario (L) and the node and link failure scenario (NL) we get eight models for which the token transfer problem will be discussed:

NA-L	NA-NL
SA-L	SA-NL
RA-L	RA-NL
GA-L	GA-NL

The goal of a token transfer algorithm is that the token, originally stored in the source, visit all fault-free nodes. Since we are working in networks with random faults, we can only require high probability of achieving this goal. This probability is called reliability of a token transfer algorithm. For every fixed $\varepsilon > 0$, a token transfer algorithm is called ε-safe if its reliability for n-node networks exceeds $1 - n^{-\varepsilon}$, for sufficiently large n.

In our probabilistic considerations we will use the following lemma known as Chernoff's bound [13].

LEMMA 2.1: *Let X be the number of successes in a series of b Bernoulli trials with success probability q. For any constant δ with $0 < \delta < 1$,*

$$\Pr (X \leq (1 - \delta)\, qb) \leq e^{-\delta^2 qb/2}.$$

The following lemma is an easy consequence of the above, directly used in the paper.

LEMMA 2.2: *Consider a series of cm Bernoulli trials with success probability $0 < r < 1$. Let $E(c, m)$ be the event that the total number of successes is at least m and let $F(c, m, k)$ be the event that in every series of k consecutive trials there is at least 1 success. Then for every $c > 4/r$,*

a) $\Pr (E(c, m)) > 1 - e^{-crm/4}$,

b) $\Pr (E(c, m) \cap F(c, m, k)) > 1 - e^{-crm/4} - cm(1 - r)^k$.

Proof: Taking $b = cm$, $q = r$ and $\delta = 1 - \frac{1}{cr}$ in Lemma 2.1 we get

$$\begin{aligned}\Pr \overline{(E(c, m))} &\leq e^{-(cr-1)^2 cmr/(2c^2 r^2)} \\ &= e^{-\left(cr-2+\frac{1}{cr}\right)\cdot m/2} < e^{-(cr-2)\cdot m/2}.\end{aligned}$$

$cr > 4$ implies $\frac{cr}{2} < cr - 2$. Hence we get

a) $\Pr (E(c, m)) > 1 - e^{-crm/4}$.

On the other hand $\Pr \overline{(F(c, m, k))} < cm(1 - r)^k$. Hence, *a)* implies *b)*. ■

3. LINK FAILURES

In this section we assume that all nodes are fault-free and links fail with probability $p < 1$. In this framework our algorithms are based on the following idea (*cf.* algorithm PATH-TRANSMISSION in [9]): arrange nodes in a line of length m and in each of cm steps attempt communication between all neighbors. If communication between a pair of neighbors succeeds with probability r in each step independently, for sufficiently large c information travels through the entire line, with high probability.

Since links fail, in order to guarantee independent transmission attempts between neighbors, we need to use distinct intermediary nodes at each time. However, unlike in the case of classical broadcasting, the token once sent by v to an intermediary x which cannot transmit it to the next node w in the line due to faulty connecting link, is not available in v to try another intermediary at the next step. Moreover, with probability p, the link xv may be faulty (although vx was fault-free) and thus the token cannot be returned to v. Thus, in the non-adaptive case, a mechanism of direct communication between intermediaries must be conceived.

Let $A = \{a_0, \ldots, a_m\}$ be a set of nodes to be visited by the token, initially situated in a_0. Let $P_i = \{v_0^i, \ldots, v_{k-1}^i\}$, for $i = 0, \ldots, m-1$, be a set of intermediaries between a_i and a_{i+1}, such that all sets $A, P_0, \ldots, P_{m-1}$ are pairwise disjoint. Let c be a positive integer constant. The nodes from A are visited by the token using the procedure NA-PATH.

procedure NA-PATH $(A, P_0, \ldots, P_{m-1})$
for $i := 1$ **to** cm **do**
  for all $j < m$ **in parallel do**
    SEND $(a_j, v^j_{i \bmod k})$
  for $s := 0$ **to** $k-1$, $s \neq i \bmod k$ **do**
    for all $j < m$ **in parallel do**
      SEND $(v^j_s, v^j_{i \bmod k})$
  for all $j < m$ **in parallel do**
    SEND $(v^j_{i \bmod k}, a_{j+1})$

The above procedure works in time $O(mk)$.

In order to describe the non-adaptive token transfer algorithm we fix two positive constants c and d. Let $k = \lceil d \log n \rceil$ and let m be the largest integer such that $m(k+1)+1 \leq n$. Let $A_1, \ldots, A_{\lceil n/m \rceil}$ be subsets of $\{1, \ldots, n\}$ such that:

$$\bigcup_{i=1}^{\lceil n/m \rceil} A_i = \{1, \ldots, n\}, \qquad A_i = \{a_0^i, \ldots, a_m^i\},$$

where a_0^1 is the source and $a_m^i = a_0^{i+1}$. Let $P_0^i, \ldots, P_{m-1}^i$ be pairwise disjoint subsets of $\{1, \ldots, n\} \setminus A_i$ of size k.

Algorithm NA-L Token Transfer
for $i := 1$ **to** $\lceil n/m \rceil$ **do**
  NA-PATH $(A_i, P_0^i, \ldots, P_{m-1}^i)$

The above algorithm works in time $O(n \log n)$.

THEOREM 3.1: *Let $p < 1$ be link failure probability and let ε be a positive constant. There exists an ε-safe non-adaptive token transfer algorithm working for n-node networks in time $O(n \log n)$.*

Proof: It is enough to show that for every $\varepsilon > 0$ there exist constants c and d such that NA-L Token Transfer is ε-safe. Fix a positive ε. Procedure NA-PATH corresponds to a Bernoulli scheme of length cm where a single trial is a transfer attempt of the token between consecutive nodes via an intermediary. A success in such a scheme has probability $r = (1-p)^2$. In order to transfer the token along the path, at least m successes are needed. Since each pair of consecutive nodes has only k intermediaries, we need to exclude the event of k consecutive failures. By Lemma 2.2 (*b*) the reliability R of NA-L Token Transfer satisfies

$$R > 1 - \left\lceil \frac{n}{m} \right\rceil (e^{-crm/4} + cm(1-r)^k)$$

for $c > 4/r$ and $r = (1-p)^2$. Thus, for sufficiently large n,

$$R > 1 - ne^{-cr \lfloor (n-1)/\lceil d \log n + 1 \rceil \rfloor /4} - (c+1)\, n\, (1-r)^{d \log n}.$$

Hence, for sufficiently large constants c and d and sufficiently large n, $R > 1 - n^{-\varepsilon}$. ■

We do not know if time complexity $O(n \log n)$ can be improved for ε-safe non-adaptive token transfer algoritms in the fault-free node scenario.

It should be noted that the non-adaptive algorithm for link and node faults scenario, presented in section 4, has the same complexity as NA-L Token Transfer. However, we chose to give the latter algorithm in the fault-free nodes case because its analysis is much simpler and the crucial path transmission idea will be used in other, more efficient algorithms later in this section.

We now turn attention to the semi-adaptive model. In this case token transfer along a path can be done more efficiently. The reason is that, since CALL transmissions are now available, it is helpful to test links joining intermediaries with nodes of the line in a preprocessing phase and then send the token only to those intermediaries which are able to send the token back in case it cannot be sent forward. In case of other intermediaries CALL transmissions are used. Thus after each step of the sending phase the token is at some node of the path. Let c, A and P_i, for $i < m$, have the same meaning as in procedure NA-PATH.

procedure SA-PATH $(A, P_0, \ldots, P_{m-1})$
for $i := 0$ **to** $k - 1$ **do** {preprocessing: connection testing}
  for all $j < m$ **in parallel do**
    CALL (v_i^j, a_j)
    if this transmission has been successful
    then a_j adds v_i^j to its GOOD_INTERMEDIARIES$_j$ list
for $i := 1$ **to** cm **do**
  for all $j < m$ **in parallel do**
    if $v^j_{i \bmod k} \in$ GOOD_INTERMEDIARIES$_j$ **then**
      SEND $(a_j, v^j_{i \bmod k})$
    else
      CALL $(a_j, v^j_{i \bmod k})$
  for all $j < m$ **in parallel do**
    SEND $(v^j_{i \bmod k}, a_{j+1})$
    SEND $(v^j_{i \bmod k}, a_j)$

The above procedure works in time $O(k + m)$.

Using the same notation as for NA-L Token Transfer we can formulate the semi-adaptive token transfer algorithm as follows.

Algorithm SA-L Token Transfer
for $i := 1$ **to** $\lceil n/m \rceil$ **do**
  SA-PATH $(A_i, P_0^i, \ldots, P_{m-1}^i)$

This algorithm works in time $O(n)$.

THEOREM 3.2: *Let $p < 1$ be link failure probability and let ε be a positive constant. There exists an ε-safe semi-adaptive token transfer algorithm working for n-node networks in time $O(n)$.*

Proof: Similarly as before it is enough to show that for every $\varepsilon > 0$ there exist constants c and d such that SA-L Token Transfer is ε-safe. The argument from the previous proof works with one modification; since the token is sent only to intermediaries that can send it back, the probability of success in the Bernoulli scheme should now be taken $r = (1 - p)^3$. ■

We finally consider adaptive models. The existence of linear time algorithm for the GA-L model is straightforward: the SA-L Token Transfer algorithm can be used. It remains to construct a corresponding algorithm for the RA-L model where only the node currently holding the token can attempt transmissions. We use previous notation.

procedure RA-PATH $(A, P_0, \ldots, P_{m-1})$
  for $i := 1$ **to** cm **do**
    for all $j < m$ **in parallel do**
      for all $v \in \{a_j\} \cup P_j \backslash \{v^j_{i \bmod k}\}$ **in parallel do**
        if the token is at v **then** SEND $(v, v^j_{i \bmod k})$
      for all $j < m$ **in parallel do**
        if the token is at $v^j_{i \bmod k}$ **then** SEND $(v^j_{i \bmod k}, a_{j+1})$

This procedure works in time $O(m)$. It should be noted that since there is only one token, the execution of the procedure does not cause multiple simultaneous transmission attempts to the same node. As before we have an algorithm working in time $O(n)$.

Algorithm RA-L Token Transfer
  for $i := 1$ **to** $\lceil n/m \rceil$ **do**
    RA-PATH $(A_i, P^i_0, \ldots, P^i_{m-1})$

The proof of the following theorem is the same as that of theorem 3.1.

THEOREM 3.3: *Let $p < 1$ be link failure probability and let ε be a positive constant. There exists an ε-safe adaptive token transfer algorithm (both in the general and restricted models) working for n-node networks in time $O(n)$.* ■

4. LINK AND NODE FAILURES

In this section we assume that links fail with probability $p < 1$, nodes other than the source fail with probability $q < 1$, all failures are independent and the source is fault-free.

We first consider the non-adaptive model. In our algorithm we will use a procedure which can be intuitively described as follows. Let $A = \{a_0, \ldots, a_{m-1}\}$ be a set of nodes to be visited by the token and let $P = \{v_0, \ldots, v_{m-1}\}$ be a set of intermediary nodes disjoint from A. Suppose that in the beginning the token is in some node of A. We think of nodes from A as points situated on a motionless circle and of nodes from P as points of a circle of equal size situated above A in such a way that, initially, v_i is straight above a_i. The second circle can turn around. In every step every node $v_i \in P$ attempts a transmission to the node a_j straight below it, then a_j attempts a transmission back to v_i and finally the upper circle makes a unit angle turn after which v_i is situated above $a_{(j+1) \bmod m}$. We perform $c_1 m$ steps for an appropriate constants c_1. Here is a formal description of this procedure.

```
procedure ROUND (A, P)
for i := 0 to c1 m - 1 do
  for all j < m in parallel do
    SEND (v_j, a_{(j+i) mod m})
    SEND (a_{(j+i) mod m}, v_j)
```

The above procedure works in time $O(m)$.

The constant c_1 will be chosen to guarantee that, for every node $a_j \in A$, at least one transmission attempt to a_j be made by a node currently holding the token, with high probability. This transmission can fail due to a faulty link; however, if the above procedure is repeated for a given set A with many pairwise disjoint sets $P_1, \ldots, P_k$ then each a_j will have many opportunities to get the token and, with high probability, one of them must succeed.

In order to guarantee that the token be in A before each execution of procedure ROUND, with high probability, we use the following procedure DROP, for an appropriate constant c_2.

```
procedure DROP (A, P)
  for i := 1 to c2 log n do
    for all j < m in parallel do
      SEND (v_j, a_{(j+i) mod m}).
```

The above procedure works in time $O(\log n)$.

Now the algorithm can be described as follows. Take a positive constant c_3. Assume for simplicity that $1 + c_3 \log n$ is an integer and divides n. Let $m = \frac{n}{1 + c_3 \log n}$. Partition the set $\{1, \ldots, n\}$ into subsets $A_0, \ldots, A_{c_3 \log n}$ of size m such that the source is in set A_0.

```
Algorithm NA-NL Token Transfer
for i := 0 to c3 log n do
  for j := 0 to c3 log n, j ≠ i do
    ROUND (A_i, A_j)
    DROP (A_i, A_j)
  DROP (A_{(i+1) mod c3 log n}, A_i).
```

The execution of the internal loop takes time $O(m \log n) = O(n)$ and hence the algorithm works in time $O(n \log n)$.

THEOREM 4.1: *Let $p < 1$ and $q < 1$ be link and node failure probabilities and let ε be a positive constant. There exists an ε-safe non-adaptive token transfer algorithm working for n-node networks in time $O(n \log n)$.*

Proof: It is enough to show that for every $\varepsilon > 0$ there exist constants c_1, c_2 and c_3 such that algorithm NA-NL Token Transfer is ε-safe. Fix a positive ε. Let c_1, c_2, c_3 be constants to be determined later. Let n be sufficiently large to satisfy $m = \dfrac{n}{1 + c_3 \log n} > \lceil c_1 \log n \rceil$. Without loss of generality assume that $\lceil c_1 \log n \rceil$ is even. Let $E_1(A, P)$ denote the event that the token visits a least $\log n$ times the set A in a fixed series of $\lceil c_1 \log n \rceil$ consecutive steps of procedure ROUND. Consider two consecutive steps i and $i + 1$ of this procedure as one Bernoulli trial. Fix nodes $v_j \in P$ and $a_{(j+i) \bmod m} \in A$ such that the token is in one of them. Steps i and $i + 1$ involve the transmissions SEND $(v_j, a_{(j+i) \bmod m})$, SEND $(a_{(j+i) \bmod m}, v_j)$, SEND $(v_j, a_{(j+i+1) \bmod m})$ and SEND $(a_{(j+i+1) \bmod m}, v_j)$. Define the success in this Bernoulli trial to be the event that nodes v_j and $a_{(j+i+1) \bmod m}$ as well as links $a_{(j+i) \bmod m}\, v_j$ and $v_j\, a_{(j+i+1) \bmod m}$ are fault-free. Thus the probability of success is $r_1 = (1 - p)^2 (1 - q)^2$. In case of success the token visits the node $a_{(j+i+1) \bmod m}$, no matter if it was previously in v_j or in $a_{(j+i) \bmod m}$. Thus obtaining at least $\log n$ successes in $\lceil c_1 \log n \rceil / 2$ Bernoulli trials (with success probability r_1) implies that event $E_1(A, P)$ holds. Hence Lemma 2.2 *a*) implies

$$\Pr \overline{(E_1(A, P))} \leq e^{-c_1 r_1 \log n / 8}$$

Procedure ROUND has at least $\left\lceil \dfrac{m}{\log n} \right\rceil \lceil c_1 \log n \rceil \geq c_1 m$ steps. If in each of $\left\lceil \dfrac{m}{\log n} \right\rceil$ groups of $\lceil c_1 \log n \rceil$ steps the token visits the set A at least $\log n$ times, the total number of visits is at least m. Thus at least one full round over A is performed. This implies that the following event holds: $E_2(A, P)$ – the event that for each node $a \in A$ the token is at least once in a node $v \in P$ currently straight above a.

It follows that

$$\Pr \overline{(E_2(A, P))} \leq e^{-c_1 r_1 \log n / 8}.$$

We also need to estimate the probability of successfully dropping the token by the procedure DROP. Let $E_3(A, P)$ be the event that upon completion of DROP(A, P) the token is in a node in A. To guarantee this it suffices that one of $c_2 \log n$ links together with the destination node be fault-free. Hence

$$\Pr \overline{(E_3(A, P))} \leq (1 - r_2)^{c_2 \log n},$$

where $r_2 = (1 - p)(1 - q)$.

Consider the algorithm NA-NL Token Transfer. Fix a node $a \in A_i$. Let $P(a)$ be the set of nodes which attempted to transmit the token to a, while currently holding it. Let $E(a)$ be the event that $|P(a)| \geq c_3 \log n$.

We have

$$\bigcap_{\substack{j \leq c_3 \log n \\ j \neq i}} E_2(A_i, A_j) \cap \bigcap_{\substack{j \leq c_3 \log n \\ j \neq i}} E_3(A_i, A_j) \cap E_3(A_i, A_{i-1}) \subset E(a).$$

The execution of DROP (A_i, A_{i-1}) guarantees the invariant that the token is in A_i when the i-th turn of the external loop starts. It follows that

$$\Pr \overline{(E(a))} \leq (c_3 \log n + 1)(me^{-c_1 r_1 \log n/8} + (1 - r_2)^{c_2 \log n}).$$

Let $F(a)$ be the event that none of the nodes from $P(a)$ succeeded in transmitting the token. Since $\Pr(F(a)) = p^{|P(a)|}$, we get

$$\begin{aligned}\Pr(F(a)) &\leq \Pr(F(a) \mid E(a)) + \Pr \overline{(E(a))} \\ &\leq p^{c_3 \log n} + (c_3 \log n + 1)(me^{-c_1 r_1 \log n/8} + (1 - r_2)^{c_2 \log n}).\end{aligned}$$

The algortihm works correctly if none of the events $F(a)$, for $a \in A$, holds. Hence its reliability satisfies the condition

$$R \geq 1 - n(p^{c_3 \log n} + (c_3 \log n + 1)(me^{-c_1 r_1 \log n/8} + (1 - r_2)^{c_2 \log n}))$$

which is larger than $1 - n^{-\varepsilon}$, for sufficiently large constants c_1, c_2, c_3. ■

The NA-NL Token Transfer algorithm can obviously be applied in all other models. In case of the model RA-NL the algorithm should be modified similarly as in section 3: nodes which do not have the token do not attempt transmissions. It follows from [7] that ε-safe token transfer cannot be accomplished in the model NA-NL or RA-NL in time $o(n \log n)$. Hence for these two models our algorithm has optimal order of time complexity. It remains to discuss the two other models: SA-NL and GA-NL.

We do not know if time complexity $O(n \log n)$ can be improved for ε-safe semi-adaptive token transfer algorithms. However we will construct an ε-safe semi-adaptive linear algorithm under an additional assumption concerning the model. This assumption consists in allowing information exchange between nodes. Namely, we suppose that the elementary procedure CALL has an additional one-bit parameter b. CALL (v, w, b) is an attempt to transmit bit b from node v to w. (It should be noted that instead of allowing one-bit information transmissions, we could equivalently allow refraining from making a CALL scheduled in a given time unit.) CALL $(v, w, -)$ means that the transmitted bit is irrelevant, it will be ignored by the destination node.

Let $A = \{a_1, \ldots, a_m\}$ be a list of nodes to be visited by the token, with $a_1 = a_m$ being the source, and let $P = \{v_1, \ldots, v_k\}$ be the set of intermediary nodes, disjoint from A. We will need $k \geq cm$ intermediaries, for an appropriate constant c. In the preprocessing phase we apply the linear time gossiping algorithm with one-bit messages, given in [10], for all nodes in the network. Upon its completion every fault-free node knows, with high probability, which nodes are faulty and which are fault-free. Let $D = \{a_{p_0}, \ldots, a_{p_{t+1}}\}$ be the sublist of A consisting of all fault-free nodes from A. ($a_{p_0} = a_{p_{t+1}}$ is the source). The token is now transfered along the following path: from a_{p_0} to the first intermediary node v_{i_0} for which links $a_{p_0} v_{i_0}$ and $v_{i_0} a_{p_1}$ are fault-free; then to node a_{p_1} and next to the first intermediary v_{i_1} such that $i_1 > i_0$, and the links $a_{p_1} v_{i_1}$ and $v_{i_1} a_{p_2}$ are fault-free, etc.

This procedure of token transfer with information exchange can be described as follows.

procedure TT-WIE (A, P)

for $i := 2$ **to** $k + m$ **do**

{each node $v \in P$ calls all nodes from A and adds to its list MARKED_v those nodes to which the call has been successful}

for all $\max(1, i - m) \leq j \leq \min(k, i - 1)$ **in parallel do**

CALL $(v_j, a_{i-j}, -)$

if this call has been successful **then**

v_j adds a_{i-j} to the list MARKED_{v_j}

for $i := 2$ **to** $k + m$ **do**

{node v_j sends 1 to a_{i-j} if the link from v to the next node in D after a_{i-j} is fault free, it sends 0 otherwise}

for all $\max(1, i - m) \leq j \leq \min(k, i - 1)$ **in parallel do**

if v_j has the token **then** SEND $(v_j,\ a_{i-j})$
else
let a_r be the first node on list D with $r > i - j$
if $a_r \in \mathrm{MARKED}_{v_j}$ **then**
CALL $(v_j,\ a_{i-j},\ 1)$
else
CALL $(v_j,\ a_{i-j},\ 0)$
if a_{i-j} got 1 from v_j in the previous time unit **then**
SEND $(a_{i-j},\ v_j)$
else
CALL $(a_{i-j},\ v_j,\ -)$

The above procedure works in time $O(k+m)$.

Let c be a constant and s the largest integer such that $s+c(s+1)+1 \leq n$. Let $A_1^*, \ldots, A_{\lceil (n-1)/s \rceil}^*$ be sets of size s such that $\bigcup_{i=1}^{\lceil (n-1)/s \rceil} A_1^* = \{2, \ldots, n\}$ and let P_i be sets of size $k = c(s+1)$ disjoint from $A_i^* \cup \{1\}$. Let A_i be the list of length $m = s+2$ whose first and last term is 1 and all other terms are elements of A_1^*. The semi-adaptive token transfer algorithm with information exchange can be written as follows.

Algorithm SA-TT-WIE
apply the linear time gossiping algorithm from [10] to diagnose all nodes.
for $i := 1$ **to** $\lceil (n-1)/s \rceil$ **do**
TT-WIE $(A_i,\ P_i)$

This algorithm works in time $O(n)$.

THEOREM 4.2: *Let $p < 1$ and $q < 1$ be link and node failure probabilities and let ε be a positive constant. There exists an ε-safe semi-adaptive token transfer algorithm with information exchange, working for n-node networks in time $O(n)$.*

Proof: Fix $\varepsilon > 0$. It follows from [10] that diagnosis of the fault status of all nodes can be done with probability of correctness exceeding $1 - n^{-\varepsilon/2}$. Assume that diagnosis has been performed correctly. Let E_i, for $i \leq \lceil (n-1)/s \rceil$ denote the event that upon completion of procedure TT-WIE $(A_i,\ P_i)$ the token visits all fault-free nodes in A_i and returns to the source. Hence E_i holds if there exists a sequence of fault-free nodes $v_{i_0}, \ldots, v_{i_t}$ in P such that $i_0 < \ldots < i_t$ and links $a_{p_j} v_{i_j}$ and $v_{i_j} a_{p_{j+1}}$, for $j = 0, \ldots, t$, are fault-free. Hence $\Pr(E_i)$ is not smaller than the probability of obtaining at least $t+1$ successes in a scheme of k Bernoulli trials with success probability $r = (1-p)^2 (1-q)$. Since $t+1 \le s+1$, Lemma 2.2 *a*) implies

$$\Pr \overline{(E_i)} \le e^{-cr(s+1)/4}.$$

Hence the reliability R of algorithm SA-TT-WIE satisfies

$$R > 1 - \frac{n^{-\varepsilon}}{2} - \left\lceil \frac{n-1}{s} \right\rceil e^{-cr(s+1)/4}$$

which is larger than $1 - n^{-\varepsilon}$ for a sufficiently large constant c. ■

We finally turn attention to the adaptive model without restrictions. In this case we will present an ε-safe token transfer algorithm working in linear time.

Let $A = \{a_1, \ldots, a_m\}$ be a set of nodes to be visited by the token and $P = \{v_1, \ldots, v_k\}$ a set of intermediary nodes disjoint from A. Assume that $1 \notin A \cup P$. We will use a procedure which can be intuitively described as follows: the token is transmitted along a DFS path in a tree of height 2 constructed in the preprocessing phase. The root of this tree is the source (node 1), vertices of level 1 are "good intermediaries": those nodes $v \in P$ for which links $1\,v$ and $v\,1$ are fault-free, and vertices of level 2 are all fault-free nodes in A. A node $v \in P$ is the parent of $a \in A$ if it is the first node $w \in P$ such that both links aw and wa are fault-free. Here is a formal description of this procedure.

procedure TOKEN-TRANSFER-IN-TREE (A, P)

{phase 1: preprocessing – tree construction}

for $i := 1$ **to** k **do** {the source finds nodes in P with which it has two way connection}

CALL $(1, v_i)$

CALL $(v_i, 1)$

if both calls successful **then**

the sources adds v_i to the list GOOD-INTERMEDIARIES

v_i sets its local flag I-AM-GOOD-INTERMEDIARY$_{v_i}$

for $i := 2$ **to** $m+k$ **do** {each node from A calls consecutive nodes from P until it finds a "good intermediary" with two way connection to it}

for all $\max(1, i-k) \le j \le \min(m, i-1)$ **in parallel do**

if the flag I-FOUND-INTERMEDIARY$_{a_j}$ is not set **then**

CALL (a_j, v_{i-j})

if (the call from a_j was successful **and** the flag

I-AM-GOOD-INTERMEDIARY$_{v_{i-j}}$ is set) **then**

CALL (v_{i-j}, a_j)

if both calls successful **then**

```
        v_{i-j} adds a_j to its list MY-NODES_{v_{i-j}}
        a_j sets its local flag I-FOUND-INTERMEDIARY_{a_j}
  {phase 2: token passing in the tree}
  for all nodes x in {1} ∪ A ∪ P in parallel do
    if x = 1 then
      for i := 1 to k do
        if v_i ∈ GOOD-INTERMEDIARIES then
          SEND (x, v_i)
          wait until token is in x
    if x ∈ P then
      wait until token is in x
      for all nodes a ∈ MY-NODES_x do
          SEND (x, a)
          wait until token is in x
      SEND (x, 1)
    if x ∈ A then
      wait until token is in x
      let v be the node from which x got the token
      SEND (x, v)
```

The above procedure works in time $O(k+m)$.

Let c be a positive constant and $k = \lceil c \log n \rceil$. Let A_1 and A_2 be subsets of $\{2, \ldots, n\}$ such that $|A_1| = |A_2| = m$, where $m = n - k - 1$, and $A_1 \cup A_2 = \{2, \ldots, n\}$. Let P_i, for $i = 1, 2$, be sets of size k, disjoint from A_i.

```
Algorithm GA-NL Token Transfer
  TOKEN-TRANSFER-IN-TREE (A_1, P_1)
  wait until 5n time units since the beginning have passed
  TOKEN-TRANSFER-IN-TREE (A_2, P_2)
```

Since execution time of the procedure TOKEN-TRANSFER-IN-TREE (A_1, P_1) may vary, the aim of waiting for $5n$ time units is that all nodes know that the first execution of the procedure has terminated (It is easy to see that $5n$ exceeds worst case execution time of this procedure, for sufficiently large n). The algorithm works in time $O(n)$.

THEOREM 4.3: *Let $p < 1$ and $q < 1$ be link and node failure probabilities and let ε be a positive constant. There exists an ε-safe adaptive token transfer algorithm (in the general model) working for n-node networks in time $O(n)$.*

Proof: It is enough to prove that for every $\varepsilon > 0$ there exists a constant c such that algorithm GA-NL Token Transfer is ε-safe. Fix $\varepsilon > 0$. Consider the procedure TOKEN-TRANSFER-IN-TREE. Let $E(a)$, for $a \in A$, be the event that for some fault-free node $v \in P$, links $1v$, va, av and $v1$ are fault-free. Thus

$$\Pr \overline{(E(a))} = (1 - (1-p)^4 (1-q))^k.$$

It follows that the reliability R of GA-NL Token Transfer satisfies

$$R > 1 - 2m(1 - (1-p)^4 (1-q))^{\lceil c \log n \rceil},$$

which exceeds $1 - n^{-\varepsilon}$, for sufficiently large c. ■

5. CONCLUSIONS

The following table summarizes our results:

Communication model / Fault model	Non-adaptive (NA)	Semi-adaptive (SA)	Adaptive	
			Restricted (RA)	General (GA)
Links fail nodes fault-free (L)	NA-L $O(n \log n)^*$	SA-L $O(n)$	RA-L $O(n)$	GA-L $O(n)$
Links and nodes fail (NL)	NA-NL $O(n \log n)$	SA-NL $O(n \log n)^*$	RA-NL $O(n \log n)$	GA-NL $O(n)$

For each of the eight models considered, time complexity of the respective token transfer algorithm appears below the name of the model in the appropriate entry of the table. All our algorithms have complexity $O(n)$ or $O(n \log n)$. Order $O(n)$ is trivially optimal. As mentioned before, order $O(n \log n)$ is also optimal for models NA-NL and RA-NL, in view of a result from [7]. Thus two problems remain open (the appropriate entries in the table are marked with a *).

Problem 1

Does there exist an ε-safe non-adaptive token transfer algorithm working in time $o(n \log n)$ if all nodes are fault-free?

Problem 2

Does there exist an ε-safe semi-adaptive token transfer algorithm without information exchange, working in time $o(n \log n)$ if both links and nodes can fail?

REFERENCES

1. K. A. Berman and M. Hawrylycz, Telephone problems with failures, *SIAM J. Alg. Disc. Meth.*, 1986, *7*, pp. 13-17.
2. D. Bienstock, Broadcasting with random faults, *Disc. Appl. Math.*, 1988, *20*, pp. 1-7.
3. S. Bitan and S. Zaks, Optimal linear broadcast, *J. of Algorithms*, 1993, *14*, pp. 288-315.
4. B. S. Chlebus, K. Diks and A. Pelc, Sparse networks supporting efficient reliable broadcasting, *Proc. of the 20th International Colloquium on Automata, Languages and Programming, ICALP-93, LNCS 700*, pp. 388-397.
5. B. S. Chlebus, K. Diks and A. Pelc, Optimal broadcasting in faulty hypercubes, *Digest of Papers, FTCS-21*, 1991, pp. 266-273.
6. B. S. Chlebus, K. Diks and A. Pelc, Fast gossiping with short unreliable messages, *Disc. Appl. Math.*, to appear.
7. B. S. Chlebus, K. Diks and A. Pelc, Waking up an anonymous faulty network from a single source, *Proc. of the 27th Annual Hawaii International Conference on System Sciences*, 1994, Vol. 2, pp. 187-193.
8. C.-T. Chou and I. S. Gopal, Linear broadcast routing, *J. of Algorithms*, 1989, *10*, pp. 490-517.
9. K. Diks and A. Pelc, Almost safe gossiping in bounded degree networks, *SIAM J. Disc. Math.*, 1992, *5*, pp. 338-344.
10. K. Diks and A. Pelc, Linear time gossiping with random faults, Rapport de Recherche RR 93/02-3, Université du Québec à Hull, 1993.
11. L. Gargano, Tighter time bounds on fault-tolerant broadcasting and gossiping, *Networks*, 1992, *22*, pp. 469-486.
12. R. W. Haddad, S. Roy and A. A. Schaffer, On gossiping with faulty telephone lines, *SIAM J. Alg. Disc. Meth.*, 1987, *8*, pp. 439-445.
13. T. Hagerup and C. Rub, A guided tour of Chernoff bounds, *Inf. Proc. Letters*, 1989/90, *33*, pp. 305-308.
14. S. M. Hedetniemi, S. T. Hedetniemi and A. L. Liestman, A survey of gossiping and broadcasting in communication networks, *Networks*, 1988, *18*, pp. 319-349.
15. E. R. Scheinerman and J. C. Wierman, Optimal and near-optimal broadcast in random graphs, *Disc. Appl. Math.*, 1989, *25*, pp. 289-297.