DISTRIBUTION OF RECORDS ON A RING OF PROCESSORS

H.L.Bodlaender and J.van Leeuwen

RUU-CS-86-6

March 1986

Rijksuniversiteit Utrecht

Vakgroep informatica

Budapestlaan 6 3584 CD Utrecht
Corr. adres: Postbus 80.012 3508 TA Utrecht
Telefoon 030-53 1454
The Netherlands

DISTRIBUTION OF RECORDS ON A RING OF PROCESSORS

H.L.Bodlaender and J.van Leeuwen

Technical Report RUU-CS-86-6
March 1986

Department of Computer Science
University of Utrecht
P.O.Box 80.012, 3508 TA Utrecht
the Netherlands

DISTRIBUTION OF RECORDS ON A RING OF PROCESSORS

H.L.Bodlaender* and J.van Leeuwen

Department of Computer Science, University of Utrecht
P.O.Box 80.012, 3508 TA Utrecht, the Netherlands

Abstract. We consider a simple version of the load-distribution problem for ring networks of n processors. Assume the networks know rounds. In each round a random processor receives a packet of information (a "record") from some external source, that must be stored somewhere on the ring. By some protocol the record is moved to a node on the ring (i.e., a processor), where the record is stored. There are no a priori constraints on the nodes where a specific record can be stored. In this paper we propose and analyse some token-based protocols for this problem that attempt to minimize the maximum number of records stored at a node (by a given total number of records) and/or the number of messages needed to achieve a fair load-distribution on the average. We also discuss how deletion of records can be incorporated in the model, while preserving the properties of the protocols.

1. Introduction. Let n processors be connected in a ring, and assume that each processor can directly communicate with its two immediate neighbours. We usually assume the ring to be oriented. The nodes and links of the network are assumed to work fault- and error free. In this paper we consider a special version of the load-distribution problem for rings of processors, modelled in the following way (cf. figure 1.1). Assume the network knows rounds. In each round the following sequence of events can/must happen:
- an arbitrary processor receives a packet of information (a "record")

---

* The work of this author was supported by the Foundation for Computer Science (SION) of the Netherlands Organization for the Advancement of Pure Research.

from some external source that must be stored somewhere on the ring,
- by some protocol the record is moved to a processor somewhere on the ring (possibly it is not moved at all),
- the record is stored at this processor, and
- (possibly) some administrative actions take place.

There are no a priori constraints on the processors where a single record can be stored. The load-distribution problem asks for protocols that attempt to minimize the maximum number of records stored at a node and/or the number of messages needed to achieve a fair load-distribution on the average. In this paper several token-based protocols will be proposed and analyzed for the problem, that are of sufficient simplicity to be useful in practice.

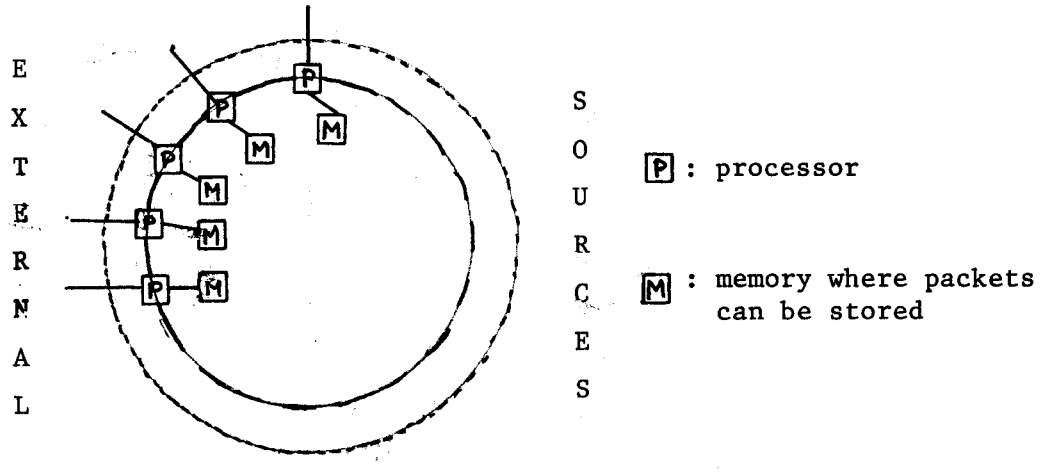


Figure 1.1. A graphical representation of the model

For the average case analysis of the protocols (viewed as communication algorithms) we assume that in each round each processor has an equal probability to receive the incoming record from an external source; i.e., for each node v and for each round the probability that v receives a record from an external source in that round is 1/n. We let p denote the total number of records stored at the nodes. Initially p will be o.

The model of the load-distribution problem as presented can be used, for instance, in the following manner: the network is used to implement a (distributed) database on which insertions of data-records and queries are performed. (In section 7 we consider deletions of records as well.) We assume that the number of updates (i.e., insertions) is relatively small over time, and the number of queries is large. Typically queries are sent around the ring and are pipelined, the bottleneck of the pipeline will be the node with the largest number of records stored in memory. Therefore it is desirable to minimize the maximum number of records stored at a node on the ring.

In this paper we will propose and analyse some token-based protocols for the given model. In section 2 we recall some elementary facts of finite Markov chain theory that are used later in the paper. In sections 3, 4 and 5 we will consider several protocols that use tokens to distribute the records in a (more or less) uniform way over the nodes. In section 6 we consider the trivial protocol that stores a record in the very node where it arrives. In section 7 we discuss how deletions can be incorporated in the model.

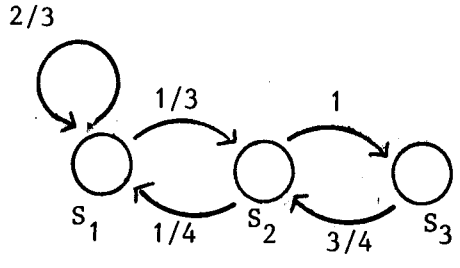
2. Elementary facts from the theory of finite Markov chains. In this section we will mention some definitions and results from the theory of finite Markov chains that are needed for the analysis in the later sections. Most of the following definitions and results, and a more detailed introduction in the theory of finite Markov chains, can be found in [2] or [4].

Consider a process S that can be in a finite number of states $s_1,\ldots,s_r$. At each step in time the process can move to another state. Further suppose that the probability that the process moves from state s_i to state s_j only depends on i and j, and on nothing else. We denote this probability by p_{ij}. This type of process is called a finite Markov chain. A finite Markov chain can be represented in two ways:

i) by a (directed) transition graph, with nodes $s_1,\ldots,s_r$ and edges from s_i to s_j labeled p_{ij} for $p_{ij} \neq o$, and
ii) by the matrix of transition probabilities

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & \\ p_{21} & p_{22} & p_{23} & \cdots & \\ p_{31} & p_{32} & p_{33} & \cdots & \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ & & & p_{rr} & \cdot \end{bmatrix}$$

For example, consider the following Markov process with 3 states s_1, s_2, and s_3. The probability that the process moves from state s_1 to state s_2 is $\frac{1}{3}$; the probability that the system stays in state s_1 is $\frac{2}{3}$; the probability that the system moves from state s_2 to state s_1 or s_3 is $\frac{1}{4}$ and $\frac{3}{4}$ respectively, and the probability that the system moves from state s_3 to state s_2 is 1 (during one step in time). This process can be represented by the transition graph from figure 2.1.a. and the matrix from figure 2.1.b.



(a)

$$\begin{bmatrix} 2/3 & 1/3 & 0 \\ 1/4 & 0 & 3/4 \\ 0 & 1 & 0 \end{bmatrix}$$

(b)

Figure 2.1.(a) A transition graph and (b) the corresponding matrix of transition probabilities

The probability that the system starts in s_j is denoted by p_j^o. For instance, if it is given that the system starts in state s_i, then $p_j^o = 0$ if $j \neq i$ and $p_i^o = 1$. The probability that the system is in state s_j

after t state-transitions from an initial state is denoted by p_j^t. It easily follows that for all j, $i \leq j \leq r$ and $t \geq o$:

$$p_j^{t+1} = \sum_{i=1}^{r} p_{ij} \cdot p_i^t \qquad (1)$$

$$\sum_{j=1}^{r} p_j^t = 1 \qquad (2)$$

$$o \leq p_j^t \leq = 1 \qquad (3)$$

Let $\vec{p}^t$ denote the rector $(p_1^t, p_2^t, \ldots, p_r^t)$, and recall that p is the matrix of transition probabilities. From equation (1) it follows that

$$\vec{p}^{t+1} = P(\vec{p}^t), \qquad (4)$$

and hence, $\vec{p}^t = P^t(\vec{p}^o)$, (5)

for all $t \geq o$.

If the transition graph of the Markov chain is strongly connected, i.e. each state can be reached from each other state by a number of transitions with non-zero probability, then the Markov chain is called ergodic. We will further only consider ergodic Markov chains.

Definition. The period of an ergodic Markov chain is the number d=gcd {l| there is a (simple) cycle of length l in the transition graph}.

Suppose the Markov chain has period d. It means that the set of states can be subdivided into d disjunct subjects $S_o, \ldots, S_{d-1}$, such that from a state in s_i $(o \leq i \leq d-1)$ only states in $s_{(i+1) \bmod d}$ can be reached by a state-transition with non-zero probability. So the system will be in a state $s_{i_o} \in S_o$, then in a state $s_{i_1} \in S_1$, then in a state $s_{i_2} \in S_2$, etc. If d=1 then the Markos chain is called regular, else it is called cyclic. For all ergodic chains it is valid that the matrix of

transition probabilities p has a unique eigenvector $\vec{p} = \langle p_1,\ldots,p_r\rangle$ with $\sum_{i=1}^{r} p_i = 1$. For this eigenrector $\vec{p}$ the following will hold:

$$0 \leq p_i \leq 1, \text{ for all } i,\ 1 \leq i \leq r \qquad (6)$$

$$P(\vec{p}) = \vec{p} \qquad (7)$$

If the chain is regular (i.e. the period d=1), then p_i denotes the asymptotic probability that the system is in state s_i after t transitions with t growing arbitrarily large.

Theorem 2.1. [2,4] If d=1, then for all j, $1 \leq j \leq r$, $\lim_{t\to\alpha} p_j^t = p_j$.

For cyclic chains a slightly weaker result holds. In this case the probabilities p_j denote the asymptotic average probability that the system is in state s_j $(1 \leq j \leq r)$, again for the number of state transitions growing arbitrarily large. (The results are also valid for regular chains.)

Theorem 2.2. [2,4] For all j, $1 \leq j \leq r$,

$$\lim_{t\to\alpha} \left(\sum_{i=0}^{t} p_j^i/t \right) = p_j \quad \text{and}$$

$$\lim_{t\to\alpha} \left(\sum_{i=0}^{d-1} p_j^{(t+i)}/d \right) = p_j, \text{ where d denotes the period of the ergodic Markov chain.}$$

It means that, in order to obtain knowledge over the average asymptotic behaviour of an ergodic finite Markov chain, it suffices to find an eigenvector $\vec{p} = \langle p_1,\ldots,p_r\rangle$ of the matrix of transition probabilities with $\sum_{i=i}^{r} p_i = 1$.

3. Token-based protocols for load-distribution with tokens going in the same direction as records. In the first, elementary protocol we propose

for the load-distribution problem, only one token is used. Initially an arbitrary processor is given the token. A record is moved forward on the ring until it arrives at the node holding the token. The record is stored at this node, and the token is passed on to the next node.

Protocol T-1

Each node v has a boolean variable token(v). Initially for exactly one node token(v) is set to true, and for every other node token(v) is set to false.

If v receives a record M either from another node or, at the beginning of a round, from an external source, then v executes:

```
if token(v) then token(v): = false;
                send a message <token> to the next node;
                store M
else send M to the next node.
endif
```

If v receives a message <token>, then v executes:

token(v): = true.

By protocol T-1 records are distributed uniformly over the nodes; each node either has $\lfloor \frac{p}{n} \rfloor$ or $\lceil \frac{p}{n} \rceil$ records stored. One easily obtains the following bounds.

Theorem 3.1. For protocol T-1 the following bounds hold:

i) The maximum number of records stored at a node is $\frac{p-1}{n} + 1$.

ii) The average difference between the maximum number of records stored at a node and p/n is $\frac{1}{2}(1 - \frac{1}{n})$.

iii) The maximum number of messages sent in a round is n.

iv) The average number of messages sent in a round is $\frac{1}{2}(n+1)$.

One might want to decrease the number of messages sent by the protocol, by allowing a larger maximum number of records stored at a processor (for a given p). However, while increasing the number of tokens in protocol T1 will indeed mainly have the effect of increasing the maximum

number of records stored at a processor, it will not help much to decrease the (average) number of messages sent by the protocol.

Protocol T-k

Each node v has a counter tokenc(v), which can assume integer values in the range o..k. The value of tokenc(v) represents the number of tokens currently held by v. The values tokenc(v) are initialized such that $\sum_{v=1}^{n}$ tokenc(v) = k, (the sum taken over all nodes v).

If v receives a record M, either from another node or, at the beginning of a round, from an external source, then v executes:

```
if tokenc(v) >o then tokenc(v): = tokenc(v)-1;
                     send a message <token> to the next node;
                     store M
else send M to the next node
endif
```

If a node v receives a message <token>, then it executes:

tokenc(v): = tokenc(v) +1.

The variables tokenc(v) are maintained so as to denote the number of tokens, present at node v. The total number of tokens will invariantly be k: after each round $\sum_{v=1}^{n}$ tokenc(v) = k. During each round exactly one processor v will execute tokenc(v): = tokenc(v)-1, and exactly one processor v' will execute tokenc(v'): = tokenc(v')+1.

Consider a certain token in some round t. The token will move to the next node in this round if one of the nodes on the path from the node containing the preceding token (this node not included) to and including the node containing the token receives a record from an outside source in this round. The probability of this event is proportional to the length of this path. (See figure 3.1.) This means that tokens will have the tendency to be at small distances from each other. (Consider for example the situation in figure 3.1. If tokens are

"close" like tokens 1 and 2 are, at distance ≥ 1, then it is more probable that the first token (i.e. token 1) will move, towards the second one, then that the second token will move away from the first one.) We will analyze this effect of token-clustering for k=2 and even n. A similar analysis is possible for odd n, and k=2. For k≥3 the same effect can be expected.

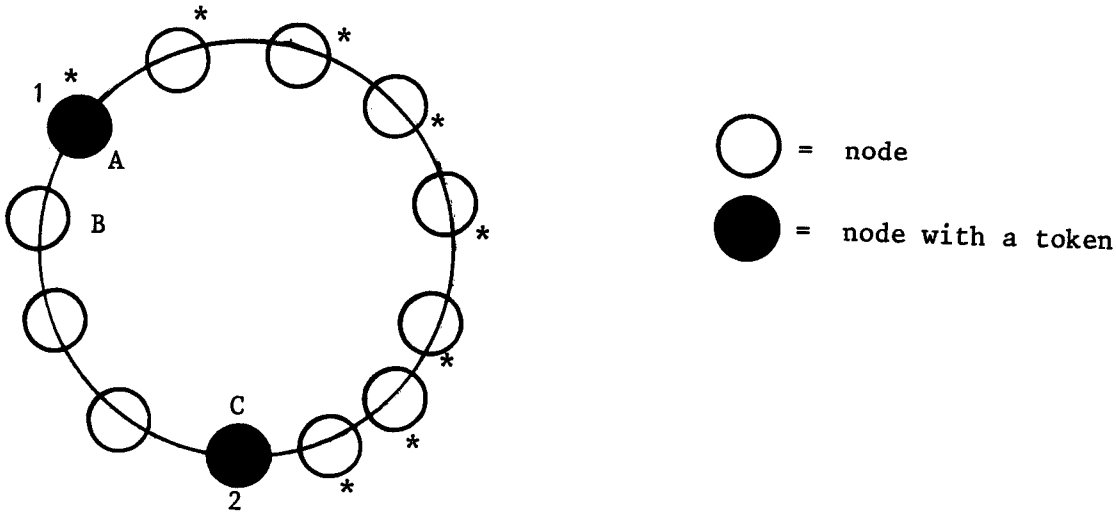


Figure 3.1. If a node with a * receives a record M from an external source, then M is stored in processor A, and token 1 moves up to B in the round, otherwise token 2 moves.

Suppose k=2, and n is even. We define the ring to be in state s_i ($0 \le i \le \frac{n}{2}$), if the shortest distance between the nodes with tokenc(v)≥1 is i. In particular, state s_o corresponds to the situation in which there is a node with tokenc(v) = 2 (meaning that the tokens reside in the same node). Denote the probability that the ring is in state s_i after t in-

sertions (i.e., for a total number of packets p=t) by p_i^t. We also assume that for all i, $o \leq i \leq \frac{n}{2}$, p_i^o is given. Naturally $\sum_{i=o}^{n/2} p_i^o = 1$ holds. With these definitions the token system on the ring is a finite Markov chain. In figure 3.2. the state-transition graph of the finite Markov chain is given.

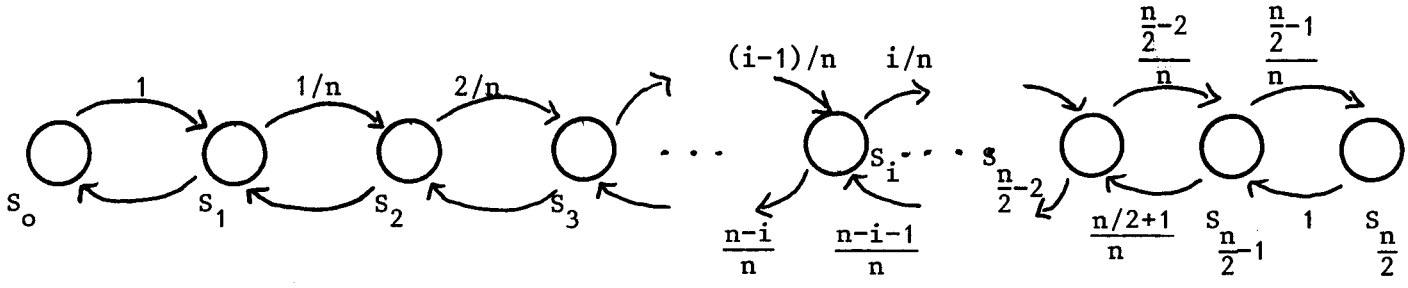


Figure 3.2. The state-transition graph.

The matrix of transition probabilities for this system is the following tridiagonal matrix A:

$$A = \begin{pmatrix}
0 & \frac{n-1}{n} & 0 & 0 & \cdots & & & & & & \cdots & 0 \\
1 & 0 & \frac{n-2}{n} & 0 & & & & & & & & \vdots \\
0 & \frac{1}{n} & 0 & \frac{n-3}{n} & & & & & & & & \\
0 & 0 & \frac{2}{n} & 0 & & & & & & & & \\
\vdots & & & & \frac{i-2}{n} & 0 & \frac{n-i}{n} & 0 & 0 \cdots & & & \\
 & & & & 0 & \frac{i-1}{n} & 0 & \frac{n-(i+1)}{n} & 0 & & & \\
 & & & & 0 & 0 & \frac{i}{n} & 0 & \frac{n-(i+2)}{n} & & & \vdots \\
 & & & & 0 & 0 & 0 & \frac{i+1}{n} & 0 & & & \\
 & & & & & & & & & 0 & \frac{\frac{1}{2}n+1}{n} & 0 \\
\vdots & & & & & & & & & \frac{\frac{1}{2}n-2}{n} & 0 & 1 \\
0 & \cdots & & & & & & & \cdots & 0 & \frac{\frac{1}{2}n-1}{n} & 0
\end{pmatrix}$$

Lemma 3.2.

i) $p_o^{t+1} = \frac{n-1}{n} p_1^t$

ii) $p_1^{t+1} = p_o^t + \frac{n-2}{n} p_2^t$

iii) $p_i^{t+1} = \frac{i-1}{n} p_{i-1}^t + \frac{n-(i+1)}{n} p_{i+1}^t \quad (2 \leq i \leq \frac{n}{2}-2)$

iv) $p_{\frac{n}{2}-1}^{t+1} = \frac{\frac{1}{2}n-2}{n} p_{\frac{n}{2}-2}^t + p_{\frac{n}{2}}^t$

v) $p_{\frac{n}{2}}^t = \frac{\frac{1}{2}n-1}{n} p_{\frac{n}{2}-1}^t$.

One easily sees that the Markov chain is ergodic (the graph of figure 3.1. is strongly connected), and that it is cyclic with period 2. We will now look for an eigenrector $\vec{p} = (p_o,\ldots,p_{\frac{n}{2}})$ of A with $\sum_{i=o}^{\frac{n}{2}} p_i = 1$.

We have the following equations for the eigenrector $\vec{p}$:

$$p_o = \frac{n-1}{n} p_1 \qquad (1)$$

$$p_1 = p_o + \frac{n-2}{n} p_2 \qquad (2)$$

$$p_i = \frac{i-1}{n} p_{i-1} + \frac{n-(i+1)}{n} p_{i+1} \quad (2 \leq i \leq \tfrac{1}{2}n-2) \qquad (3)$$

$$p_{\frac{1}{2}n-1} = \frac{\frac{1}{2}n-2}{n} p_{\frac{1}{2}n-2} + p_{\frac{1}{2}n} \qquad (4)$$

$$p_{\frac{1}{2}n} = \frac{\frac{1}{2}n-1}{n} p_{\frac{1}{2}n-1} \qquad (5)$$

We will estimate p_o and p_1.

Lemma 3.3. For every i with $2 \leq i \leq \frac{n}{2}-1$, $p_i = \frac{i-1}{n-1} p_{i-1}$ and (hence) $p_i = \frac{1}{\binom{n-2}{i-1}} p_1$.

Proof. Use induction on i. For i=2 the lemma follows directly :

$$p_1 = p_o + \frac{n-2}{n} p_2 \text{ and } p_o = \frac{n-1}{n} p_1 \Rightarrow \frac{n-2}{n} p_2 = \frac{1}{n} p_1 \Rightarrow p_2 = \frac{1}{n-2} p_1 .$$

Now suppose the lemma holds for a certain i, $2 \le i \le \frac{n}{2} - 1$. Then $p_i = \frac{i-1}{n} p_{i-1} + \frac{n-(i+1)}{n} p_{i+1}$ and $p_i = \frac{i-1}{n-i} p_{i-1} \Rightarrow \frac{n-(i+1)}{n} = p_i - \frac{i-1}{n} \cdot \frac{n-i}{i-1}$ $p_i = \frac{1}{n} p_i \Rightarrow p_{i+1} = \frac{i}{n-(i+1)} p_i$, and the induction step is complete. The second expression for p_i follows likewise. □

Lemma 3.4. For every i,j with $3 \le i \le j \le \frac{n}{2}$, $p_i \ge p_j$.

Proof. The result follows from lemma 3.3. and equation (5). □

Lemma 3.5.

i) $p_1 = \frac{1}{2} + O\left(\frac{1}{n^2}\right)$.

ii) $p_o = \left(\frac{n-1}{n}\right) \cdot \frac{1}{2} + O\left(\frac{1}{n^2}\right)$.

Proof.

i) $$1 = \sum_{i=o}^{\frac{1}{2}n} p_i = \frac{n-1}{n} p_1 + p_1 + \frac{1}{n-2} p_1 + \frac{1}{\binom{n-2}{2}} p_1 + \sum_{i=4}^{\frac{1}{2}n} p_i$$

$$\le \left(2 - \frac{1}{n} + \frac{1}{n-2} + \frac{2}{(n-2)(n-3)} \right) p_1 + \left(\tfrac{1}{2} n - 3 \right) \cdot p_4 \Big)$$

$$= \left(2 - \frac{2}{n(n-2)} + \frac{2}{(n-2)(n-3)} + \left(\tfrac{1}{2} n - 3\right) \cdot \frac{6}{(n-2)(n-3)(n-4)} \right) p_1$$

$$= \left(2 + O\left(\frac{1}{n^2}\right) \right) p_1, \text{ hence } p_1 = \frac{1}{2 + O\left(\frac{1}{n^2}\right)} = \frac{1}{2} + O\left(\frac{1}{n^2}\right) .$$

ii) Use (i) and equation (1). □

Lemma 3.5. shows that $p_o + p_1$ (= the probability that the two tokens are in the same node (p_o) or in neighbouring nodes (p_1)) grows to 1 as n and p/n become arbitrarily large. For instance, if n=20, then one can show (as in lemma 3.3.) that $p_1 > 0.4950$ and $p_o > 0.4702$. It means that in the case that n=20, in the long run, on the average, at least 96% of

the time the distance between the two tokens is o or 1. This clearly proves the effect of "token-clustering".

Theorem 3.6. Let n be even. Assume in protocol T-2 that the tokens start in opposite nodes, i.e. $p^{o}_{\frac{n}{2}} = 1$. The following bounds are valid for protocol T-2.

i) The maximum number of records stored at a node, in worst-case, is $\frac{p}{n} + 1\frac{1}{2} - \frac{2}{n}$.

ii) The average difference between the maximum number of records stored at a node and p/n approaches $1\frac{1}{4}$, as n and p/n tend to infinity.

iii) The worst-case number of messages sent in a round is n.

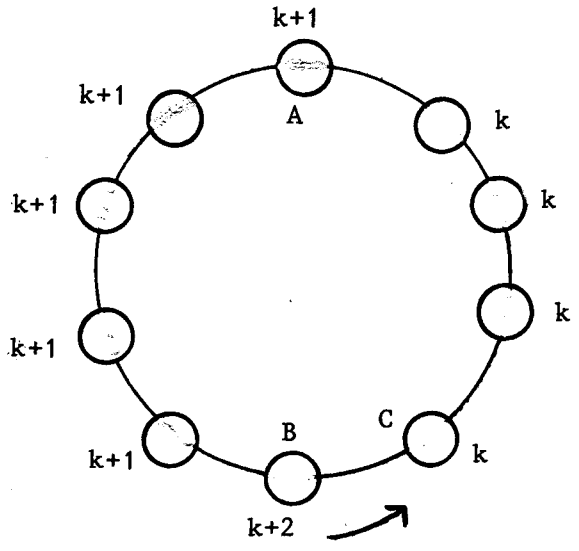
iv) The average number of messages sent in a round approaches $\frac{1}{2}n + O(1)$ as n and p/n tend to infinity.

Proof.

i) The maximum arises when both tokens are in the same node, that is a node, neighbouring a node where a token started in round o. Then $\frac{n}{2}$ nodes have k records stored, $\frac{n}{2}-1$ nodes have k+1 records stored and one node (the node where the tokens reside) has k+2 records stored, for certain constant k. (cf. figure 3.2.). Now $p = \frac{n}{2}\cdot k + (\frac{n}{2}-1)(k+1) + k+2$, and hence the maximum number of records stored at a node is $k+2 = \frac{p}{n} + 1\frac{1}{2} - \frac{2}{n}$. With some straightforward analysis one obtains that no other case gives a larger maximum.

ii) Notice that for our analysis we may assume that the system stays in the set of states $\langle s_o, s_1 \rangle$. Suppose the system is in state p_o and the distance between a node where a token started and the node where the tokens reside is i, with $o \leq i \leq \frac{n}{2}$. With an analysis similar to (i), one obtains that the maximum number of records stored at a node is $\frac{p}{n} + 1 + \frac{i}{n} + O(\frac{1}{n})$. A similar bound can be obtained for the case that the system is in state p_1. As i will be uniformly distributed over the range $\langle 0, 1, \ldots, \frac{n}{2}-1 \rangle$, the average difference between the maximum number of records stored at a node and $\frac{p}{n}$ will approach $1 + \sum_{i=o}^{\frac{1}{2}n-1} \frac{i}{n} + O(\frac{1}{4}) = 1\frac{1}{4} + O(\frac{1}{n})$.

iii) The worst-case number of messages n is sent in the case that the



A: node where token 1 started

B: node where token 2 started

C: node where both tokens reside

Each number denotes the number of records stored at the corresponding node.

Figure 3.2. The maximum number of records stored at a node with protocol T-2.

system is in state s_o, and the record arrives in the node following the node where the two tokens reside. The record must make (n-1) steps until it arrives in a node with a token; one extra message is needed to send the token.

iv) Note that the average number of messages needed in a round with the system in state s_i is $\frac{1}{n}\left(\sum_{j=1}^{i} j + \sum_{j=1}^{n-i} j\right)$ $= \frac{1}{n}\left(\frac{1}{2}(i+1)i+\frac{1}{2}(n-i)(n-i+1)\right) = \frac{1}{2}n+\frac{i^2}{n}-i+\frac{1}{2}$. As n and p/n tend to infinity, the probability that the system is in state p_o approaches $\frac{1}{2}$, the probability that the system is in state p_1 approaches $\frac{1}{2}$, and the probability that the system is in a state p_i with $i \geq 2$ approaches o. So the average number of messages will approach $\frac{1}{2}\left(\frac{1}{2}n+\frac{1}{2}\right)+\frac{1}{2}\left(\frac{1}{2}n+\frac{1}{n}-1+\frac{1}{2}\right)=\frac{1}{2}n+\frac{1}{2n}$. □

The case that n is odd can be analyzed similarly, and yields similar results. Theorem 3.6. indicates that in order to decrease the average number of messages sent in a round, other approaches to fair load-distribution must be followed.

4. Token-based protocols for load-distribution with tokens moving in the opposite direction as packets. The effect of token clustering that arises in protocol T-k ($k \geq 2$) can be avoided in the following, simple manner: let tokens move in the opposite direction as packets. Protocols T'-1 and T'-k are obtained from T-1 and T-k by replacing the statement "send a message <token> to the next node" by "send a message <token> to the preceding node". It is easily seen that for protocol T'-1 exactly the same bounds hold as for protocol T-1 (cf. theorem 3.1.).

The main result of this section concerns the asymptotic analysis of the average performance of the protocols T'-k for $k \geq 2$, with finite Markov chain theory. The protocols appear to have a fairly ideal behaviour. Not only the effect of token-clustering is avoided, but tokens that have a small distance to each other will have the tendency of increasing the distance to each other. The analysis shows that the protocols T'-k indeed decrease the (asymptotic) average number of messages required for load-distribution, while the maximum number of records stored at a node (by a given p) shows only a very small increase.

Suppose the records travel in counter-clockwise direction on the ring. Thus, tokens are sent in clockwise direction according to the instructions of protocol T'-k. Number the tokens consequtively, in counter-clockwise direction (i.e. the direction in which records travel), from 1 to k. Let a_i^t denote the distance of token i to token (i+1) after t rounds, measured along the ring in the same direction, a_k^t denotes the distance of token k to token 1. Note that always $\sum_{i=1}^{k} a_i^t = n$. If, after the t'th round, $a_i^t = a_i$ for $1 \leq i \leq k$ and certain a_i $(1 \leq i \leq k)$ with $\sum_{i=1}^{k} a_i = n$, then the system is said to be in state $S_{a_1 \ldots a_k}$.

The probability that the system is in state $S_{a_1 \ldots a_k}$ after t rounds is denoted by $p^t_{a_1 \ldots a_k}$.

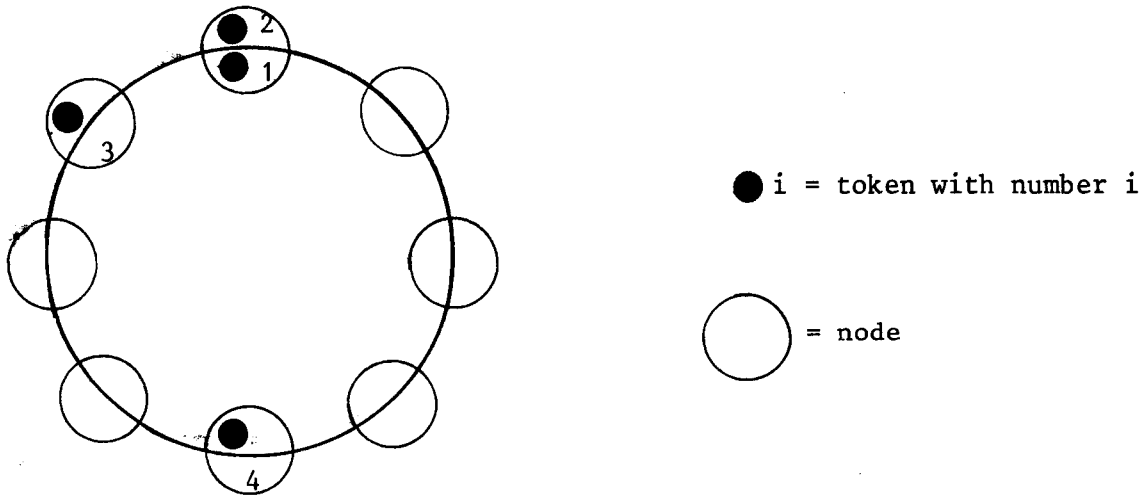


Figure 4.1. A system in state $S_{0,1,2,5}$. (k=4, n=8).

If the system is in state $S_{a_1 \ldots a_k}$ with $a_i \geq 1$, then the (i+1)st token can move if one of the nodes on the path from the node, next to the node containing the i'th token, to the node containing the (i+1)st token receives a record M in this round from an external source. The probability of this event is precisely $\frac{a_i}{n}$. As a result of the event $a_{i+1}^{t+1} = a_{i+1}^{t}+1$ and $a_i^{t+1} = a_i^{t}-1$, and the resulting state is $S_{a_1 \ldots a_{i-1} a_{i+1}+1 \ldots a_k}$. (If i=k, then $s_{a_1 \ldots a_i - 1 a_{i+1}+1 \ldots a_k}$ denotes $s_{a_1+1-\, a_2 \ldots a_{k-1} a_{k-1}}$.)

Theorem 4.1.

i) For all $a_1, \ldots a_k \geq o$ with $\sum_{i=1}^{k} a_i = n$ and $t \geq o$,

$$p^{t+1}_{a_1 \ldots a_k} = \sum_{\substack{1 \leq i \leq k \\ a_{i+1} \geq 1}} p^{t}_{a_1 \ldots a_i+1 a_{i+1}-1 \ldots a_k} \cdot \frac{a_{i+1}}{n} .$$

ii) For all $t \geq 0$, $\sum p^t_{a_1,\ldots a_k} = 1$, where the summation is taken over all $a_1,\ldots,a_k \geq 0$ with $\sum_{i=1}^{k} a_i = n$.

The system of tokens is seen to be a finite Markov chain, which is ergodic and has period k. For the asymptotic average behaviour of the system we have to solve the following system of equations characterizing an eigenvector of the matrix of transition probabilities as required:

(*)

$$- p_{a_1\ldots a_k} = \sum_{1\leq i\leq k \atop a_{i+1}(\geq 1} p_{a_1\ldots a_i + 1 a_{i+1} - 1\ldots a_k} \cdot \frac{a_{i+1}}{n}$$

for all $a_1,\ldots,a_k \geq 0$ with $\sum_{i=1}^{k} a_i = n$ (1)

$$- \sum p_{a_1\ldots a_k} = 1,$$ where the summation is taken over all $a_1,\ldots,a_k \geq 0$ with $\sum_{i=1}^{k} a_i = n$. (2)

From Markov chain theory it follows that this system of equations will have exactly one solution, and the $p_{a_1\ldots a_k}$ characterized by the system of equations denote the asymptotic average probability that the system is in state $S_{a_1\ldots a_k}$.

Theorem 4.2. A solution to the system of equations (*) is:

$$p_{a_1\ldots a_k} = \frac{n!}{a_1!\ldots a_k!} \frac{1}{k^n},$$

for all $a_1,\ldots,a_k \geq 0$ with $\sum_{i=1}^{k} a_i = n$.

Proof. Use the well-known fact that $\sum_{0\leq a_1\ldots a_k \atop \sum_{i=1}^{k} a_i = n} \frac{n!}{a_1!\ldots a_k!} = k^n$.

Now equation (2) follows immediately. Further we have for all $a_1,\ldots,a_k \geq 0$ with $\sum_{i=1}^{k} a_i = n$

$$\sum_{\substack{1\leq i\leq k \\ a_{i+1}\geq 1}} \frac{n!}{a_1!\ldots(a_i+1)!(a_{i+1}-1)!\ldots a_k!} \frac{a_i+1}{n} =$$

$$\sum_{i=1}^{k} \frac{n!}{a_1!\ldots a_i!\, a_{i+1}!\ldots a_k!} \frac{a_{i+1}}{n} = \frac{n!}{a_1!\ldots a_k!} \cdot \sum_{i=1}^{n} \frac{a_{i+1}}{n} = \frac{n!}{a_1!\ldots a_k!} \cdot \square$$

Theorem 4.3. The average number of steps a record must go according to protocol T'-k until it arrives in a node with a token is equal to $\frac{1}{2}k(n-1)$.

Proof. If the system is in state $s_{a_1\ldots a_k}$, then the average number of steps a record must go (in that round) until it arrives in a node with a token is

$$\frac{1}{n} \sum_{i=1}^{k} \sum_{j=0}^{a_i-1} j = \sum_{i=1}^{k} \frac{a_i(a_i-1)}{2n} .$$

So the (total) average number of steps a record must go is equal to:

$$\left\{ \sum_{\substack{a_1\ldots a_k\geq 0 \\ \sum_{i=1}^{k} a_i = n}} \right\} p_{a_1\ldots a_k} \cdot \sum_{j=1}^{k} \frac{a_j(a_j-1)}{2n} =$$

$$\left\{ \sum_{\substack{a_1\ldots a_k\geq 0 \\ \sum_{i=1}^{k} a_i = n}} \right\} \frac{n!}{a_1!\ldots a_k!} \cdot \frac{1}{k^n} \cdot \sum_{j=1}^{k} \frac{a_j(a_j-1)}{2n} =$$

$$\sum_{j=1}^{k} \left\{ \sum_{\substack{a_1\ldots a_k\geq 0 \\ \sum_{i=1}^{k} a_i = n}} \right\} \frac{n!}{a_1\ldots a_k!} \cdot \frac{1}{k^n} \frac{a_j(a_j-1)}{2n} =$$

$$k \cdot \left\{ \begin{array}{l} \sum \\ a_1 \ldots a_k \geq 0 \\ \sum_{i=1}^{k} a_i = n \end{array} \right\} \frac{n!}{a_1! \ldots a_k!} \cdot \frac{1}{k^n} \cdot \frac{a_1(a_1-1)}{2n} =$$

$$\left\{ \begin{array}{l} \sum \\ a_1 \geq 2 \\ a_2 \ldots a_k \geq 0 \\ \sum_{i=1}^{k} a_i = n \end{array} \right\} \frac{(n-2)!}{(a_1-2)!a_2! \ldots a_k!} \cdot \frac{1}{k^{n-1}} \cdot \frac{n-1}{2} =$$

$$\left(\left\{ \begin{array}{l} \sum \\ a_1 \ldots a_k \geq 0 \\ \sum_{i=1}^{k} a_i = n-2 \end{array} \right\} \frac{(n-2)!}{a_1! \ldots a_k!} \right) \cdot \frac{1}{k^{n-1}} \cdot \frac{n-1}{2} =$$

$$k^{n-2} \cdot \frac{1}{k^{n-1}} \cdot \frac{n-1}{2} = \frac{1}{2k}(n-1). \quad \square$$

Consider theorem 4.3. for the situation that $k|n$ and the system is in state $S_{\frac{n}{k}\frac{n}{k}\ldots\frac{n}{k}}$, i.e., the tokens are equally spread out over the ring at distance $\frac{n}{k}$. In this "best possible case", the average number of steps a record has to go until it arrives in a node with a token is $\frac{1}{2}(\frac{n}{k}-1)$, which differs less than $\frac{1}{2}$ (!) from the calculated asymptotic average bound for the algorithm T'-k.

Theorem 4.4. Assume the system starts in a state $s_{a_1 \ldots a_k}$, with $a_i \in \{\lceil \frac{n}{k} \rceil, \lfloor \frac{n}{k} \rfloor\}$ for all i, $1 \leq i \leq k$. The following bounds are valid for protocol T'-k:

i) the worst-case maximum number of records stored at a node is $\frac{p}{n} + \frac{1}{2}k + O(1)$.

ii) the worst-case number of messages sent in a round is n.

iii) the (asymptotic) average number of messages in a round is $\frac{1}{2k}(n-1) + 1$. (n fixed and p tending to infinity).

The assumption in theorem 4.4. is only necessary for (i). As we start

in a balanced state, it means that also for small round number t, the average number of messages sent in a round will be close to $\frac{1}{2k}(n-1) + 1$. It is currently open to give a good estimate for the (asymptotic) average difference between $\frac{p}{n}$ and the maximum number of records stored at a node.

5. Token-based protocols for load-distribution with tokens going in both directions. In this section we will briefly consider protocols for the load-distribution problem with two types of tokens: tokens that travel in the same direction as the records, and tokens that travel in the opposite direction. We will only analyse the situation with one token of each type. To obtain the best possible bounds, we assume that the tokens start in neighbouring nodes, as in figure 5.1.

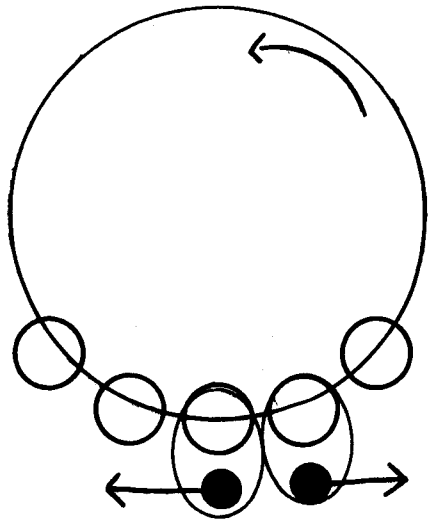


Figure 5.1. Tokens start in neighbouring nodes and go in opposite directions.

Protocol TT'.

Each node v has boolean variables left(v), right(v). Initially there is exactly one node v with left(v) = true, all other nodes have left(v) =

false, the right neighbour of this node has right(v) = true, all other nodes have right(v) = false.

If v receives a record M, either from another node or, at the beginning of a round, from an external source, then v executes:

```
if left(v) then     left(v):=false;
                    send <left> to left neighbour;
                    store M
elif right(v) then  right(v):=false;
                    send <right> to right neighbour;
                    store M
else send M to right neighbour
endif.
```

If v receives a message <left>, then it executes:

left(v):=true.

If v receives a message <right>, then it executes:

right(v):=true.

Lemma 5.1. After the t'th round the distance of the "left"-token to the "right"-token is (t+1) mod n (measured in the direction of the packets).

Proof. Use induction on t. For the distance between the tokens it is unimportant whether the "left"-token moves or the "right"-token moves. □

Theorem 5.2. The following bounds are valid for protocol TT':

i) The worst-case maximum number of records stored at a node is $\frac{p}{n} + (1-\frac{1}{n})$.

ii) The average difference between this maximum and $\frac{p}{n}$ is $\frac{1}{2}(1 - \frac{1}{n})$.

iii) The worst-case number of messages sent in a round is n.

iv) The average number of messages sent in a round is $\frac{1}{3}n + \frac{2}{3n} + \frac{1}{2}$, in the long run.

Proof.

i),ii) Each node has either $\lceil \frac{p}{n} \rceil$ or $\lfloor \frac{p}{n} \rfloor$ packets stored.

iii) Trivial.

iv) If the distance of the left token to the right token is i, then on the average a record must go $\frac{i(i-1)}{2n} + \frac{(n-i)(n-i-1)}{2n}$ steps, in that round, until it arrives in a node with a token. If i=o then the average distance is $\frac{1}{2}(n-1)$. From lemma 5.1. we have that each of the distances between the tokens o,1,...,n-1 appears equally often (in the long run). So the total average number of steps needed until a record arrives in a node with a token is

$$\frac{1}{n} \sum_{i=o}^{n-1} \left(\frac{i(i-1)}{2n} + \frac{n-i}{2n} (n-i-1) \right) = \frac{1}{n} \left(\sum_{i=o}^{n} \left(\frac{i \cdot (i-1)}{n} \right) - \tfrac{1}{2}(n-1) \right)$$

$$= \frac{n(n+1)(2n+1)}{6n^2} - 1 + \frac{1}{2n} = \frac{1}{3}n + \frac{2}{3n} - \tfrac{1}{2} .$$

One extra message is needed to move the token. □

Note that each of the bounds for protocol TT' are equal to or better than the corresponding bounds for protocol T-1 or T'-1; the average number of messages is smaller by a factor of about 1.5.

6. Load-distribution by "direct placing". In this section we will examine the trivial protocol that just stores a record in the node that receives a record from an external source.

This simple protocol uses no messages; the worst-case number of records stored at a node is p (if packets always arrive in the same node). Thus, the main problem in analyzing this protocol is the expected maximum number of packets stored at a node (for a given p). We can reformulate the problem as follows:

> Suppose we have n urns and p balls. For every ball an urn is chosen at random (with for each urn a probability 1/n that it is chosen). What is the expected number of balls that is in the urn with the largest number of balls?

We denote this expected number by E_n^p. We know of results on related problems (see e.g. [1], [2] or [3]), but we do not know of results that give estimates or even exact expressions for E_n^p. For even p one can derive an exact expression for E_2^p (i.e., the case of two urns).

Theorem 6.1. For p even, $E_2^p = \frac{p}{2} + \binom{p}{\frac{1}{2}p} \cdot \frac{p}{2^{p+1}}$.

Proof. The probability that the first urn contains i balls and the second urn contains p-i balls is $\binom{p}{i}/2^p$. So

$$E_2^p = \sum_{i=o}^{p} \left(\frac{\binom{p}{i}}{2^p} \max(i,p-i) \right) = 2 \cdot \sum_{i=\frac{1}{2}p+1}^{p} \frac{\binom{p}{i} i}{2^p} + \frac{\binom{p}{\frac{1}{2}p} \cdot \frac{1}{2}p}{2^p}$$

$$= 2 \cdot \sum_{i=\frac{1}{2}p+1}^{p} \binom{p-1}{i-1} \frac{p}{2^p} + \frac{p}{2^{p+1}} \binom{p}{\frac{1}{2}p}$$

$$= \sum_{i=o}^{p-1} \binom{p-1}{i} \frac{p}{2^p} + \frac{p}{2^{p+1}} \binom{p}{\frac{1}{2}p} = \frac{1}{2}p + \binom{p}{\frac{1}{2}p} \cdot \frac{p}{2^{p+1}}. \quad \square$$

The following results enable us to obtain estimates for E_n^p for all p and n.

Theorem 6.2. Let $k|n$. Then $E_k^p \leq \frac{n}{k} E_n^p$.

Proof. Simulate the experiment with k urns in the following way: First carry out the experiment with n urns (and p balls). Then we add together the balls of the first $\frac{n}{k}$ urns, the balls of the next $\frac{n}{k}$ urns, etc. So urn i in the final experiment contains the balls of urns $\frac{n}{k}(i-1)+1$ to $\frac{n}{k} \cdot i$ of the first experiment. Now each ball has a probability $\frac{1}{k}$ to be in the i'th urn, for all i, $1 \leq i \leq k$; and the choices of urn for the balls are independent of each other. So this is a correct way to simulate the "E_k^p-experiment".

If the maximum number of balls in an urn in the experiment with n urns is m, then the maximum number of balls in an urn in the experiment with k urns is at most $\frac{n}{k} \cdot m$. So $E_k^p \leq \frac{n}{k} E_n^p$. $\square$

Theorem 6.3. Let n be even. Then

$$E_2^p \geq E_n^p + \frac{p-E_n^p}{n-1}\left(\tfrac{1}{2}n-1\right) .$$

Proof. We simulate the experiment with 2 urns as in the proof of theorem 6.2. First carry out the experiment with n urns and p balls. Then choose $\frac{n}{2}$ urns at random, and add the balls of these urns together. Also add the balls of the remaining urns together. Again each ball has probabilities $\frac{1}{2}$ to be in the first, or in the second set, respectively, and the choices for the individual balls do not depend on each other. So this is a correct simulation of the experiment with 2 urns and p balls.

Let the balls be numbered 1...p. There is a unique urn X (after the first experiment), with the following properties:

- no urn contains more balls than X does
- no urn that contains the same number of balls as X, contains a ball with a higher number than the highest numbered ball in urn X.

(We need the last constraint to have a unique urn X which contains the maximum number of balls.) Let X contain m balls. The number of balls in the set where X belongs to, averaged over all possible choices of $\frac{n}{2}$ urns is $m + \frac{p-m}{n-1}\left(\frac{1}{2}n-1\right)$.
(Note that each of the $\frac{1}{2}n-1$ urns $\neq$ X in the set contains on the average $\frac{p-m}{n-1}$ balls). Hence $E_2^p \geq E_n^p + \frac{p-E_2^p}{n-1}\left(\frac{1}{2}n-1\right)$. □

Lemma 6.4. $E_n^p + \frac{1}{n} \leq E_n^{p+1} \leq E_n^p + 1$.

Proof. First do the E_n^p-experiment, and then add the (p+1)'st ball. The probability that this ball is in an urn with the maximum number of balls (so the ball increases the maximum by 1) is at least $\frac{1}{n}$. The maximum will never be increased by more than 1. □

Definition. For all p, n $\geq$ 1, let $X_n^p = E_n^p - \frac{p}{n}$.

Theorem 6.5. Let n $\geq$ 2. Then

$$\frac{n-1}{n} X_{n+1}^p \leq X_n^p \leq \frac{n+1}{n} X_{n-1}^p$$

Proof. The experiment with n urns and p balls is simulated as follows: do the 'E^p_{n+1}-experiment', and then take urn n+1 and choose for each of its balls randomly one of the first n urns (each urn with probability 1/n). The expected total number of balls in an urn now again is E^p_n.

Number the balls 1...p. Again there is a unique urn X with the following properties:

- no urn contains more balls than X does,
- no urn that contains the same number of balls as X, contains a ball with a higher number than the highest numbered ball in urn X, with regard to the outcome of the first experiment (with n+1 urns).

The probability that X is one of the first n urns is $\frac{n}{n+1}$. Suppose it is one of the first n urns. The expected number of balls in X is E^p_{n+1}. The expected number of balls in urn (n+1) now is $\frac{p-E^p_{n+1}}{n}$. So, the expected number of extra balls, added to urn X in the last stage of the experiment is $\frac{p-E^p_{n+1}}{n^2}$. This means that the expected number of balls in urn X after the last stage of the experiment is $E^p_{n+1} + \frac{p-E^p_{n+1}}{n^2}$.

If urn X is urn (n+1), then note that the maximum number of balls in an urn after the last stage of the experiment is at least $\frac{p}{n}$.

$$\text{So } E^p_n \geq \frac{1}{n+1}\left(\frac{p}{n}\right) + \frac{n}{n+1}\left(E^p_{n+1} + \frac{p-E^p_{n+1}}{n^2}\right)$$

$$= \frac{1}{n+1}\frac{p}{n} + \frac{n}{n+1}\left(\frac{p}{n+1} + X^p_{n+1} + \frac{p-\frac{p}{n+1} - X^p_{n+1}}{n^2}\right) = \frac{1}{n+1}$$

$$= \frac{p}{n} + \frac{n-1}{n}\cdot X^p_{n+1}, \text{ and hence } X^p_n \geq \frac{n-1}{n} X^p_{n+1}.$$

Similarly $X^p_n \leq \frac{n+1}{n} X^p_{n+1}$. □

Theorem 6.6. Let p be even.

i) If n is even, then

$$\frac{p}{n} + \frac{1}{n}\frac{p}{2^p}\binom{p}{\frac{1}{2}p} \leq E^p_n \leq \frac{p}{n} + \frac{n-1}{n}\frac{p}{2^p}\binom{p}{\frac{1}{2}p}$$

ii) If n is odd, then

$$\frac{p}{n} + \frac{1}{n+1}\,\frac{p}{2^p}\binom{p}{\frac{1}{2}p} \leq E_n^p \leq \frac{p}{n} + \left(1-\frac{1}{n}2\right)\frac{p}{2^p}\binom{p}{\frac{1}{2}p}.$$

Proof.

i) From theorem 6.3. one derives (by elementary formulamanipulation) that $X_n^p \leq 2\,\frac{n-1}{n}\,X_2^p$. The desired inequality now follows from this fact and theorems 6.1 and 6.2.

ii) Use the previous result and theorem 6.5. □

For odd p, an estimate of E_n^p follows directly from lemma 6.4. Also, note that by using Stirling's approximation formula:

$$e^{-\frac{1}{3}p}\frac{\sqrt{2}}{\sqrt{\Pi}}\sqrt{p} \leq \frac{p}{2^p}\binom{p}{\frac{1}{2}p} \leq \frac{\sqrt{2}}{\sqrt{\Pi}}\sqrt{p}.$$

Theorem 6.6. shows for the trivial protocol that the maximum number of records stored at a node will be $\frac{p}{n} + O(\sqrt{p})$, on the average.

7. Load-distribution under insertions and deletions. An interesting question is how deletions can be handled. To model this situation we assume that at the beginning of a round two different types of record can arrive: records that have to be inserted (stored at a node), and records that have to be deleted (removed from the node where it was stored earlier). Clearly, for a deletion, one always needs to find the node where the record is stored first: this costs n-1 messages in the worst-case and $\frac{1}{2}n$ messages in the average case.

As before we let p denote the total number of records that are presently stored on the ring, and we let q denote the number of records that were inserted, but deleted later (and thus are no longer stored). The total number of insertions that has taken place thus equals p+q.

We will consider two approaches to deletions. First we consider the trivial protocol that handles a deletion by removing a record from the node where it is stored, and does nothing further. Next we consider a protocol that also moves records from one node to another node, in order to maintain a (more or less) uniform distribution of the records over the nodes. For the average case analysis we assume that all records are equally likely to be deleted.

First assume the trivial deletion protocol is used in combination with the trivial insertion protocol of section 6. It is obvious that records that are inserted and later deleted do not influence the number of records stored at any node after the deletion of the record. So the maximum number of records stored in a node will be E_n^p, which can be estimated as in section 6.

Next we consider the case where the trivial deletion protocol is used in combination with one of the protocols of sections 3, 4 and 5 (protocols T-1, T-k, T'-1, T'-k or TT'). For this case we observe the following. Consider the experiment of section 6, and let F_n^p denote the expected minimum number of balls in an urn (when n urns and p balls are used in the experiment). Let f(p,n) denote the average difference between p/n and the maximum number of packets at a node with the insertion algorithm that is used, if there were only insertions. We fix a moment t in time, and again let p denote the number of stored records and q the number of records that were once inserted, but deleted later.

Lemma 7.1. The expected maximum number of records stored at a node is at most

$$\frac{p+q}{n} + f(p+q,n) - F_n^q .$$

Proof. The expected maximum number of records that are inserted at a node (so are stored presently or were stored previously and were deleted thereafter) equals $\frac{p+q}{n} + f(p+q,n)$. If for each node the probability that a record is deleted from that node is equal, then the expected minimum number of records deleted from that node is equal, then the expected mimimum number of records deleted from a node is F_n^q. However, this probability is proportional to the number of records stored at a node, so nodes with a larger number of stored records will have a larger expected number of deleted records. Note that this will result in an expected maximum number of records which will be (slightly) smaller than $\frac{p+q}{n} + f(p+q,n) - F_n^q$. □

F_n^q can be analysed similar to E_n^q. We write $F_n^q = \frac{p}{n} - Y_n^p$.

Theorem 7.2.

i) Let p be even. $F_2^p = \frac{p}{2} - \frac{p}{2^{p+1}} \binom{p}{\frac{1}{2}p}$.

ii) Let $k|n$. $F_k^p \geq \frac{n}{k} F_n^p$.

iii) Let n be even. $F_2^p \leq F_n^p + \frac{p-E_n^p}{n-1} (\frac{1}{2}n-1)$.

iv) Let $n \geq 2$. $\frac{n-1}{n} Y_{n+1}^p \geq Y_n^p \geq \frac{n+1}{n} Y_{n-1}^p$.

v) $F_n^p + \frac{1}{n} \leq F_n^{p+1} \leq F_n^p + 1$.

Proof. i) Use $F_2^p + E_2^p = p$.
ii), iii), iv), v) Similar to the analysis in section 6. □

Theorem 7.3. Let p be even.

i) If n is even then $\frac{p}{n} - \frac{n-1}{n} \frac{p}{2^p} \binom{p}{\frac{1}{2}p} \leq F_n^p \leq \frac{p}{n} - \frac{1}{n} \frac{p}{2^p} \binom{p}{\frac{1}{2}p}$

ii) If n is odd then $\frac{p}{n} - (1 - \frac{1}{n}2) \frac{p}{2^p} \binom{p}{\frac{1}{2}p} \leq F_n^p \leq \frac{p}{n} - \frac{1}{n+1} \frac{p}{2^p} \binom{p}{\frac{1}{2}p}$.

For odd p, theorem 7.2.(v) shows how to obtain extimates for F_n^p with theorem 7.3. Again we note that $e^{-\frac{1}{3}p} \frac{\sqrt{2}}{\sqrt{\Pi}} \sqrt{p} \leq \frac{p}{2^p} \binom{p}{\frac{1}{2}p} \leq \frac{\sqrt{2}}{\sqrt{\Pi}} \sqrt{p}$. Now we have the following result.

Theorem 7.4. The expected maximum number of records stored at a node is at most $\frac{p}{n} + f(p+q,n) + \frac{\sqrt{2}}{\sqrt{\Pi}} \cdot \sqrt{q}$.

Proof. Use lemma 7.1.
$\frac{p+q}{n} + f(p+q,n) - F_n^q \leq \frac{p+q}{n} + f(p+q,n) - (\frac{q}{n} - \frac{q}{2^q} \binom{q}{\frac{1}{2}q}) \leq \frac{p}{n} + f(p+q,n) + \frac{\sqrt{2}}{\sqrt{\Pi}} \sqrt{q}$.
□

Finally we consider a load-distribution protocol that tries to maintain a balanced distribution of the records over the nodes after each deletion by moving a record from some node to the node where the

record was deleted (in general).

The protocol is used in combination with a protocol T'-k, for some $k \geq 1$. The tokens used for insertions in algorithm T'-k are also used for deletions. For a deletion of a record M the algorithm proceeds as follows. First the node v is found where the record M is stored, and must be deleted. Then we find the first node preceding v that contains a token. This token is moved to the next node (i.e., in the direction in which records move under insertions). From this (next) node, a record M' is deleted, M' is moved to the node v where M was stored and M' is inserted at v.

Protocol DT-k.

Each node v has a variable tokenc(v), as in protocol T-k, and a boolean variable gap(v), which is initually false for all nodes v. For insertions protocol T-k is used.

If v receives a message <delete, M>, either from another node or, at the beginning of a round, from an external source, then v executes:

```
if M is stored at v then delete M,
                         gap(v):=true:
                         send <find token> to preceding node
else send <delete, M> to next node
endif.
```

If v receives a message <find token>, then v executes:

```
if tokenc(v) > o then tokenc(v):= tokenc(v)-1;
                      send <collect> to next node
else send <find token> to preceding node
endif.
```

If v receives a message <collect>, then v executes:

```
tokenc(v):=tokenc(v) +1;
if gap(v) then gap(v):=false
else delete a packet M' from v
       send <fillgap, M'> to next node
endif.
```

If v receives a message <fillgap, M'>, then v executes:

```
if gap(v) then store M' at v;
                gap(v):=false
else send <fillgap, M'> to next node
endif.
```

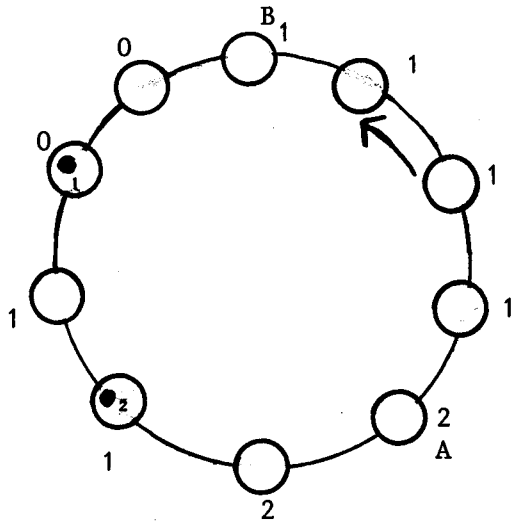
The following properties are invariants for the protocol after each completion of a round:
- gap(v)=false for all nodes v,
- the total number of tokens is k ($\sum$ tokenc(v)=k, where the sum is taken over all nodes v.)

However, to prove the protocol correct we need to show that always a packet M' exists when a collect message is generated. To show this, follow individual tokens as insertions and deletions take place. In each round, one of the tokens moves to a neighbouring node. For each node v, let $\Psi(v)$ be the number of tokens for which v lies on the path, starting with the node after the token, upto and including the node where the token initially started, in the direction in which insertion records travel. An example, with 2 tokens is given in figure 7.1.

Lemma 7.5. The following property is invariant under algorithm DT-k after each completion of a round:
- there is a $j \geq -k$ such that each node v has exactly $\Psi(v)+j$ records stored.

Proof. First we note the property is initially true: note that $\Psi(v)=k$ for all nodes v.

If in a round t a record is inserted in a node v, then there are two cases:



A: node where token 1 started initially

B: node where token 2 started initially

i: token with number i

Figure 7.1. An illustration of the function $\Psi(v)$. With each node v the number $\Psi(v)$ is shown. The arrow shows the direction in which insertion records move.

- v is not the node where the (moving) token initially started. Then the path, starting with the node after the token to the node where the token initially started, is extended by the node v, hence $\Psi(v)$ increases by 1, and for all other nodes w $\Psi(w)$ does not change. The invariant is again valid after the completion of the round.
- v is the node where the moving token initially started. Then the path contains after the round only the node v, where formerly each node on the ring was on the path (from the node after the token to the node where the token initially started.) So $\Psi(w)$ drops by 1 for all nodes $w \neq v$, but $\Psi(v)$ does not change. By increasing j by 1, one sees that the invariant is again valid after completion at the round.

A similar analysis can be given for deletions. □

It follows that if a node v does not contain a token, then it has at least the same number of records stored as the next node. This shows that if a node v receives a <collect> message and not(gap(v)) holds, then there is at least one record stored at v (so there exists a record M' that can be deleted from v.) Without much difficulty one obtains the

following result.

Theorem 7.6. Assume the system starts in a state $S_{a_1 \ldots a_k}$ with for all i, $1 \leq i \leq k$, $a_i \in \{\lceil \frac{n}{k} \rceil, \lfloor \frac{n}{k} \rfloor\}$, i.e. the distance between each pair of 'neighbouring' tokens is $\lceil \frac{n}{k} \rceil$ or $\lfloor \frac{n}{k} \rfloor$. The following bounds are valid for protocol DT-k:

i) The worst-case maximum number of records stored at a node is $\frac{p}{n} + \frac{1}{2}k + O(1)$.

ii) The worst-case number of messages sent in a round with an insertion is n.

iii) the worst-case number of messages sent in a round with a deletion is 3n-1.

The average number of messages needed in a round of protocol DT-k can be estimated in the following way. Suppose that in a certain round t a record is deleted. The probability that a certain token i moves in that round (to the next node) is proportional to the total number of records stored at the nodes on the path from the node with token i to the node, immediately before the node with token i+1. (So if this total number of records is p', then the probability is p'/p.) The numbers $\Psi(v)$ range between o and k, so if p/n becomes large, the probability becomes approximately proportional to the length of the path between the tokens. If we assume that this probability is exactly the length of this path divided by n, then the resulting model can be analyzed similar as in section 4. For instance, one can easily derive, that - given that in round t+1 a deletion occurs - for all $a_1 \ldots a_k \geq o$

$$p^{t+1}_{a_1 \ldots a_k} = \sum_{\substack{1 \leq i \leq k \\ a_i \geq 1}} p^{t}_{a_1 \ldots a_i - 1 a_{i+1} + 1 \ldots a_k} \cdot \frac{a_{i+1} + 1}{n}.$$

As in section 4 one can derive that the average probabilities that the system is in state $S_{a_1 \ldots a_k}$ approach $\frac{n!}{a_1! \ldots a_k!} \cdot \frac{1}{k^n}$, if the number of insertions (and deletions) becomes arbitrarily large. If follows that an insertion costs approximately n/k +O(1) messages on the average, and a deletion costs $\frac{1}{2}n + \frac{2n}{k} + O(1)$ messages on the average.

(In our analysis we always assumed that one never tries to delete records that are not stored in some node.)

References.

[1] David, F.N. and D.E. Barton, Combinatorial chance, Charles Griffin & Co., London, 1962.

[2] Feller, W., An introduction to probability theory and its applications, Vol.1, J. Wiley & Sons, New York, 1968.

[3] Johnson, N.L. and S. Kotz, Urn models and their applications, J. Wiley & Sons, New York, 1977.

[4] Kemeny, J.G. and J.L. Snell, Finite Markov Chains, D. van Nostrand Co., Princeton, New Jersey, 1960.