KERNFORSCHUNGSZENTRUM

KARLSRUHE

Februar 1974 KFK 1734

Institut für Datenverarbeitung in der Technik

Multiple Copy Files in Computer Networks

E. Holler

GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.

KARLSRUHE

Als Manuskript vervielfältigt

Für diesen Bericht behalten wir uns alle Rechte vor

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 1734

Institut für Datenverarbeitung
in der Technik

Multiple copy files in computer networks

Elmar Holler

Gesellschaft für Kernforschung mbH., Karlsruhe

Abstract

Multiple copy files in computer networks

Depending on the ratio of update to query traffic for a file in a computer network, a multiple copy file allocation may be more economical than a single copy allocation.

For symmetrical and homogeneous computer networks an upper limit for the ratio of update to query traffic is given, for which the multi-node assignment of a file represents an economic solution.

In addition, a decentralized control mechanism for synchronizing the updating of a file with multiple copies is shown. The mechanism guarantees consistency of the information stored in different copies of a file by ensuring that updating requests are executed in the same order at all copies of the file. Identical queries issued between two consecutive updatings will yield the same result, regardless of the copy the use.

This paper is a revised version of a presentation given at the European Workshop on Computer Networks in Arles, April 24 - May 4 th, 1973.

Zusammenfassung

Mehrfach realisierte Dateien in Rechnernetzen

Die Verteilung mehrerer Kopien einer Datei über ein Netz von Rechnern kann ökonomischer sein als die Einrichtung einer einzigen Kopie. Die optimale Verteilung und Zahl der Kopien einer Datei hängt vom Verhältnis Änderungsverkehr/Abfrageverkehr zwischen den Rechnern und der Datei ab.

Für symmetrische, homogene Rechnernetze wird die obere Schranke des Verhältnisses Änderungsverkehr/Abfrageverkehr angegeben, für die die Mehrfachrealisierung einer Datei noch ökonomisch vertretbar ist.

Zusätzlich wird ein Überwachungsmechanismus skizziert, der zur Koordination der Änderungen mehrfach realisierter Dateien unter Wahrung der Konsistenz der abgespeicherten Information eingesetzt werden kann.

Der Bericht ist die überarbeitete Fassung eines Vortrages, der anläßlich des 1st European Workshop on Computer Networks in Arles im Mai 1973 gehalten wurde.

1. Introduction

One of the main reasons for implementations of (resource sharing) computer networks is the principal capability of such networks to provide for efficient use of common data- and program files by establishing files as shared resources. The main problems arising upon design and implementation of computer nets with shared files are: Incompatibility (data-representation, data structures), economy (cost for data transmission and data storage), maintenance (updating of files), etc.

This paper is concerned with two of the problems mentioned above:

- How to find the optimal (most economic) allocation of copies of files in a computer network.
- How to synchronize updating of files with multiple copies.

The optimal allocation may result in the allocation of more than one copy of the same file in the network, in which case the affected file constitutes a resource distributed over the entire net. This situation will not affect any query-process concerning the file, but for update processes a synchronizing concept has to be developed in order to guarantee the integrity of the file as one (logical) resource in cases where concurrent updates (originating from different nodes in the net) may occur.

To illustrate the updating problem in the case of a file with multiple copies, consider the following example (cf. Fig. 1).

Given a simple 3 node network, where two copies of a file are allocated to nodes labeled 1 and 2 resp. Let us assume that requests for updating the file (including both copies) are generated at node 1 and node 2 simultaneously. If no synchronizing mechanism is implemented, nodes 1 and 2 will start immediately to update their own copies while mutual update requests are sent over the communication system to update the other copy (see(a) of fig. 1). Upon arrival of the requests at the other nodes, the local updates are already in progress (or finished), due to the transmission delays (see(b) in fig. 1). At the end of the whole updating procedure (see (c) in fig. 1) a query process (e.g. q_3) may find different versions of the file depending on wether node 1 or node 2 is accessed (e.g. in the case where the same record of the file was changed twice).

We proceed now to show by means of a simple cost model, under which conditions it is reasonable to allocate more than one copy of a file in a computer network, and how an optimal allocation of copies of files can be determined. This model is related to the mathematical models used for describing the allocation of resources in networks in order to evaluate e.g. the most economic locations for warehouses, concentrator locations in communication nets etc. [1], [2], [3]. Similar models were first applied to file allocation problems by Chu [4], Whitney [5] and Casey [6]..

Finally we demonstrate a deadlock free synchronizing concept which can be applied to synchronize updatings in file systems with mutiple copy files [7].

2. File Allocation

Given a computer network with n nodes (computing systems) and m different files. Let X_{ij} denote whether a copy of file j is located at node i or not

$$X_{ij} = \begin{cases} 1 & \text{file j stored at node i} \\ 0 & \text{else} \end{cases}$$

$i = 1\ (1)\ n \qquad j = 1\ (1)\ m$

The number of copies of file j is given by

$$r_j = \sum_i X_{ij}$$

To simplify the model, the cost for transmitting a unit of information via the communication system from node i to node k is expressed by d_{ik}, the fixed cost for storing a copy of file j at node k per unit time by σ_{kj} with

$$\sigma_{kj} = L_j \, S_{kj}$$

where L_j denotes the "length" of file j in units of information and S_{kj} the storage cost per unit of information and unit time for file j at node k. The volume of query and update traffic per unit time from node i to file j ist represented by λ_{ij} and ψ_{ij} respectively.

The total cost (per unit time) for the file system is then given by:

$$(1) \quad Z = \sum_{j=1}^{m} \left[\underbrace{\sum_{i,k=1}^{n} \psi_{ij} d_{ik} X_{kj}}_{\text{update cost}} + \underbrace{\sum_{i=1}^{n} \lambda_{ij} g_i(I_j)}_{\text{query cost}} + \underbrace{\sum_{k=1}^{n} \sigma_{kj} X_{kj}}_{\text{storage cost}} \right] = \sum_j C_j$$

where the value of $g_i(I_j)$ for a given indexcombination ij depends on the strategy by which copies of files are accessed for query-requests. If e.g. the copy with minimal transmission cost is selected, then (cf. [6])

$$(2) \qquad g_i(I_j) = \min_{k \varepsilon I_j} \; d_{ik}$$

If the copies are selected at random (uniformly distributed), except if the copy is located at the same node, we obtain

$$\text{(2a)} \quad g_i (I_j) = \sum_{k=1}^{n} d_{ik} \quad X_{kj} \quad (1-X_{ij})/r_j \qquad \text{(cf. [4])}$$

I_j denotes the index set with $I_j = \{k | X_{kj}=1\}$

The problem of optimal file allocation in the network is solved by manipulating the X_{ij} such that (1) is a minimum, thus becoming a nonlinear zero-one programming problem.

Chu [4] has investigated a special case of this optimization problem, introducing the following constraints:

a) the number of copies r_j is known for j=1(1) m

b) there is a limited storage-capacity b_i at each node i:

$$\sum_j X_{ij} L_j \leq b_i \qquad i= 1(1)\ n$$

c) the time required to retrieve file j for the i-th computer may not exceed a given limit T_{ij}

Chu shows that this nonlinear zero-one programming problem can be reduced to a linear zero-one programming problem for which a solution can be obtained (cf. [4]).

We shall focus our attention to some useful properties of the cost function (1) in cases where the numbers of copies of the files are not known in advance and the allocations are not subject to constraints (b) and (c) (i.e. the allocations of files are mutually independant). The minimum of

$$Z = \sum_j C_j$$

is obtained by minimizing the C_j individually. For minimizing procedures applicable in this case cf. [6].

Let ρ be the ratio of update to query traffic for a given file in a computer network. Then we can write the cost function C (file index j deleted), assuming $\psi_i/\lambda_i = \rho$ at all nodes.

$$(3) \quad C = \rho \sum_{k=1}^{n} X_k \sum_{i=1}^{n} \lambda_i d_{ik} + \sum_{i=1}^{n} \lambda_i g_i(I) + \sum_{k=1}^{n} \sigma_k X_k$$

For a single node assignment (file assigned to node k) we obtain

$$(3a) \quad C^k = (1+\rho) \sum_{i=1}^{n} \lambda_i d_{ik} + \sigma_k \qquad \text{if } g_i(I) \text{ is given by (2).}$$

If C^1 is the optimal 1-node assignment, then

$$(4) \quad C^k = C^1 + \alpha_k \quad , \quad \alpha_k \geq 0$$

and, using the above, we can write

$$(5) \quad C - C^1 = (r\rho-\rho-1)\left[\sum_{i=1}^{n} \lambda_i d_{i1} + \sigma_1/(1+\rho)\right] + \sum_{i=1}^{n} \lambda_i g_i(I) + \left(\sum_{k=1}^{n} (\rho\alpha_k + \sigma_k) X_k\right)/(1+\rho)$$

As can be seen from (5) any r-node assignment of a file is more costly than the optimal on-node assignment if

$$(6) \quad r\rho - \rho - 1 \geq 0$$

which implies

$$\rho = \psi_i/\lambda_i \geq 1/(r-1) \quad \text{for all } i$$ (see [6])

It follows out of (6) that, if the volume of update traffic approaches the volume of query traffic at all nodes, the optimal file assignment is a one-node assignment.

In order to obtain the maximum allowable (economical) limit ρ**max** for the ratio update/query traffic for a given multinode file allocation I with r copies we set

(7) $$C - C^1 = 0$$

thus obtaining an equation for ρmax which can be solved in the case of a symmetrical homogeneous net with $\alpha_k = 0$ for all k and

(8) $$\sigma_i = \sigma_1 \text{ for } i = 1(1)n, \quad \sum_{k=1}^{n} \sigma_k X_k = r\ \sigma_1$$

(equal storage cost for the file copies at all nodes in the net). Symmetry of a computer network with respect to file allocation is achieved in a homogeneous net if

(9) $$\sum_{i=1}^{n} \lambda_i\ d_{ik} = \text{const} \quad , \quad k = 1(1)\ n$$

for all single node assignments (file assigned to any node k).

Examples of symmetrical nets are given in Fig. 2.
Nets (a) and (b) show the desired behaviour if λ_i is constant for all i and the transmission costs between all adjacent nodes are equal. Net (c) with the central switch is equivalent to (b).

As a solution of (7) we get

(10) $$\rho max = (1-c)/(r-1) - d \quad \text{for } r = 2(1)\ n$$

where

(11) $$c = (\sum_i \lambda_i g_i(I)) / \sum_i \lambda_i d_{i1} = Q_I/Q_1$$

is the ratio r-node allocation query cost/1-node allocation query cost and

(12) $$d = \sigma_1 / \sum_i \lambda_i\ d_{i1} = \sigma_1 / Q_1$$

is the ratio storage cost/query cost for the 1-node allocation.

Since $\rho \geq 0$ in all cases, we have the necessary condition

(13) $$0 \leq d \leq (1-c)/(r-1) \leq 1$$

Using (11) and (12) we can write instead of (10)

$$(14) \qquad \rho max = \frac{Q_1 - Q_I - (r-1)\ \sigma_1}{(r-1) Q_1}$$

Thus $\rho max > 0$ if the savings of query cost $Q_1 - Q_I$ achieved by a multinode allocation exceeds the increase in storage cost $(r-1) \cdot \sigma_1$ caused by the multinode allocation.

To illustrate the qualitative behaviour of ρmax as a function of the number of r optimally allocated copies we focus our attention again to the examples in Fig. 2.
Table 1 shows the values of the query-cost Q_I as a function of r (optimal allocation) for examples (a) and (b) if $\lambda_1 = 1$ for all nodes and $d_{ik} = 1$ between adjacent nodes.

Table 1

r \ Q_I	net (a)	net (b)
1	9	4
2	4	3
3	3	2
4	2	1
5	1	0
6	0	-

It follows from Table 1 that in the case of net (a) c may be approximated by 1/r, yielding

$$0 \leq \rho \leq \frac{1}{r} - d$$

For net (b) with

$$c = 1-(r-1)/(n-1)$$

we get

$$0 \leq \rho \leq \frac{1}{n-1} - d$$

i.e. in that special case ρmax depends only on the maximum number of nodes in the net and on the ratio storage cost/query cost of the single node assignment, but not on r (for r≥2).

The investigations described above show at least for the homogeneous and symmetrical net, that, whenever the ratio update/query traffic is smaller than the limit given by (10) an r-node allocation of a file is (economically) justified.

3. Multiple Copy Updating

As shown above, the most economic solution of the file allocation problem may result in an allocation of more than one copy of a file. There may be, of course, additional reasons for multiple copy file allocations as e.g. system availability and reliability, response time etc.

In most cases it is an inalienable requirement that, between two consecutive updates of a file, the contents of all of its copies are identical, thus granting that simultaneous queries yield the same result regardless of the copies they use.

In order to comply with this requirement, every multiple copy updating control system has to perform the following functions:

a) accept and synchronize updating requests
b) inhibit queries
c) lock copies as soon as queries in progress are finished
d) update copies
e) unlock copies

Clearly the transition from a) to c) may cause an interlock for concurrent updates if no synchronization is taken care of during stage a).

The synchronization of update requests asks for a complete knowledge of the global state of the file system in question. This can be easily achieved by means of a centralized structure of the file control system, an example of which is shown in Fig. 3. Incoming update requests can be processed on a first come - first served basis in this case.

The disadvantages of central file control are abvious:

- if the file control is affected by any failure the total file system may become inoperative
- even local queries have to be announced to the central file control.

The alternative to centralized file control is decentralized or distributed file control, where each node in the net has its own file manager, being responsible for access coordination and-control for updatings and queries concerning the associated copies of files. To provide for update synchronization in a network of computers with distributed file control the following concept is proposed (cf. [7]).

Let (Π, D, M) describe a file system, where the set of processes

$$\Pi = \{\Pi_1, \ldots., \Pi_r\}$$

is the set of file managers associated with those r nodes of a network where the r copies of a file are allocated. The file managers may be considered as processes performing the functions a) to d) described above.

$$D = \{P, Q, R\}$$

denotes the set of file manager states corresponding to the assumed states of the total file. These states may be regarded as reusable resources, consisting of r units each, which may be requestet and assigned to the processes in Π:

P => file (with all of its copies) prepared for update
Q => preparation confirmed for total file
R => file allocated for updating

Finally

$M = \{M_{EXT}, M_{PR}, M_{CF}\}$

describes a set of messages of the following format:

M_i =

A	B	C

where A contains information concerning the message type identifying the type of demand represented by the message:

demand type:=

EXT	external request for update
PR	demand to prepare for an updating request initiated by the file manager specified in B
CF	demand to confirm preparation for the updating request initiated by the file manager specified in B

The contents of B and C identify the initiating file manager of the request the message refers to and the file manager sending the message respectively. The initiating file manager is assumed to determine the request priority in an unambiguous manner.

Messages are produced and consumed by the file manager processes and shall therefore be called consumable resources.

With the above assumptions, we are able to interpret the file system as a general resource system (cf. [8]), the states of which may be represented by means of a graph (general resource graph), the vertices or nodes being the processes, reusable and consumable resources. Request edges are directed from a process node,assignment edges are edges directed from reusable resource nodes,and producer edges are edges directed from consumable resource nodes (for details cf. [8]).

If the file managers are designed as shown in the diagram in Fig. 4 (queries still in progress are neglected) the file system constitutes a general resource system with the following properties:

- the processes request only one resource unit at a time
- whenever a consumable resource (message) is requested by a process (file manager) the process is blocked
- whenever reusable resource $D_i \varepsilon D$ is requested the corresponding file manager process is blocked
- there will never be a process node Π_i in the general resource graph which is not a sink or has no path directed to a sink, since in the case of any process Π_i requesting a consumable resource (thus being blocked), there will be always another process Π_j, $j \neq i$, which is a producer of the required resource and is not blocked. Reusable resources are only requested, when available.

As shown by Holt [8] the properties described above are necessary and sufficient conditions for a process in a general resource system not being in a deadlock.

To illustrate the representation of a multiple copy file system by means of a general resource graph, an example is shown in Fig. 5 for r = 3.

4. Conclusion

The problem of multiple copy file allocation in computer networks has been investigated in order to show the economic feasible upper limit of the number of copies allocated as a function of the proportion of update traffic to query traffic generated in the network. For the special case of the homogeneous and symmetrical net the functional relationship between the ratio of update to query traffic and the number of copies was given.

After having shown, that a multiple copy file allocation may be reasonable from the economical point of view, the concept of a multiple copy updating mechanism was demonstrated and examined for its deadlock prevention properties.

References

[1] E. Feldman et al.
Warehouse location under continuous economies of scale
Management Science, Vol. 12, No. 9, pp.67o-684, May 1966

[2] K. Spielberg
Algorithm for the simple plant location problem with some side conditions
Operations Research, Vol. 17, 1969, pp. 85-111

[3] W.D. Frazer
An approximate algorithm for plant location under piecewise linear concave costs
IBM Research Report RC 1875, July 1967

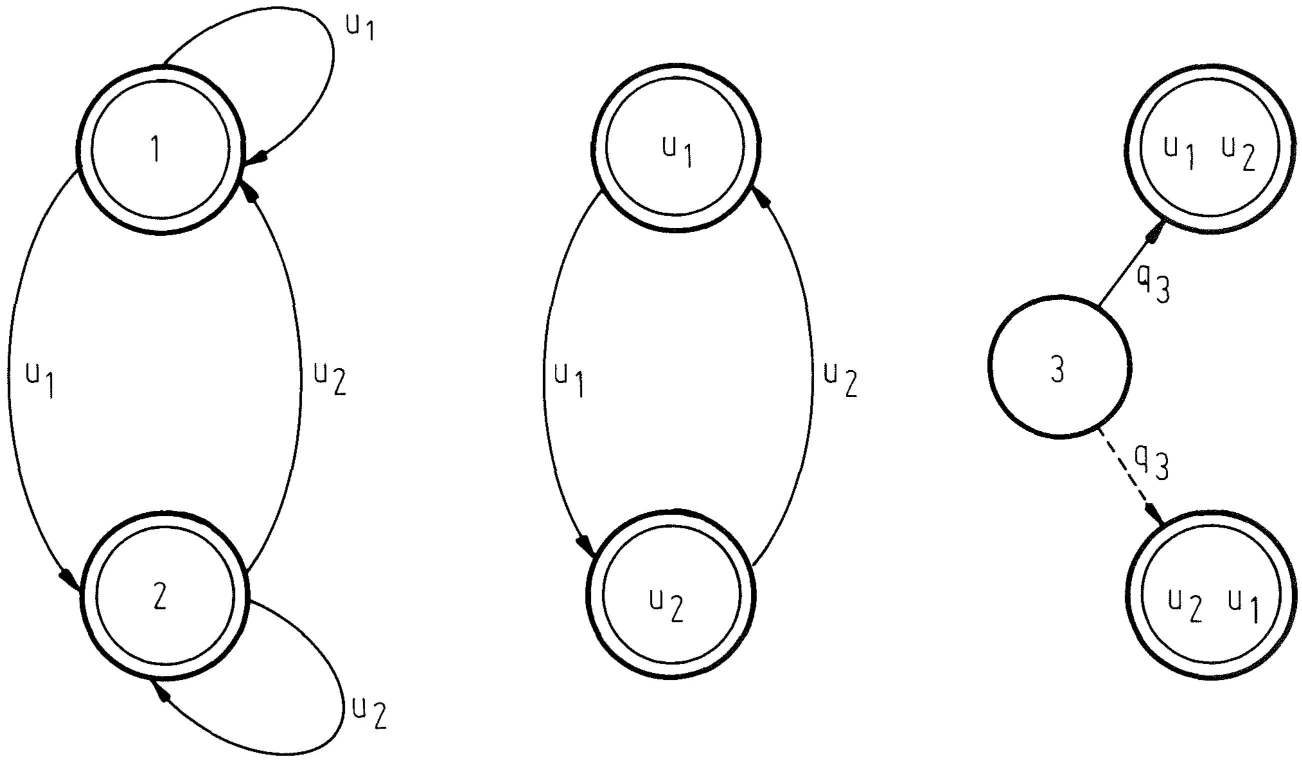
[4] W.W. Chu
Optimal file allocation in a multiple computer system
IEEE Trans. on Computers, Vol. C18, No. 10, 1969

[5] V.K.M. Whitney
A study of optimal file assignment and communication network configuration in remote-access computer message processing and communication systems
SEL Technical Report No. 48
Dept. of Electr. Engrg. University of Michigan, Sept. 1970
(PhD Dissertation)

[6] R.G. Casey
Allocation of copies of a file in an information network
AFIPS Proc. Spring Joint Comp. Conf.1 1972

[7] E. Merten
Zugriffssynchronisation bei mehrfach geführten Dateien in Rechnernetzen. Diplomarbeit
Universität Karlsruhe, März 1973

[8] R. Holt
Some deadlock properties of computer systems
Computing Surveys, Vol. 4, No. 3, Sept. 1972



a) requests for updating u_1, u_2

b) local copies updated

c) all copies updated and query from node 3 (q_3)

Fig 1 Asynchroneous updating of a 2-copy file in a 3-node net

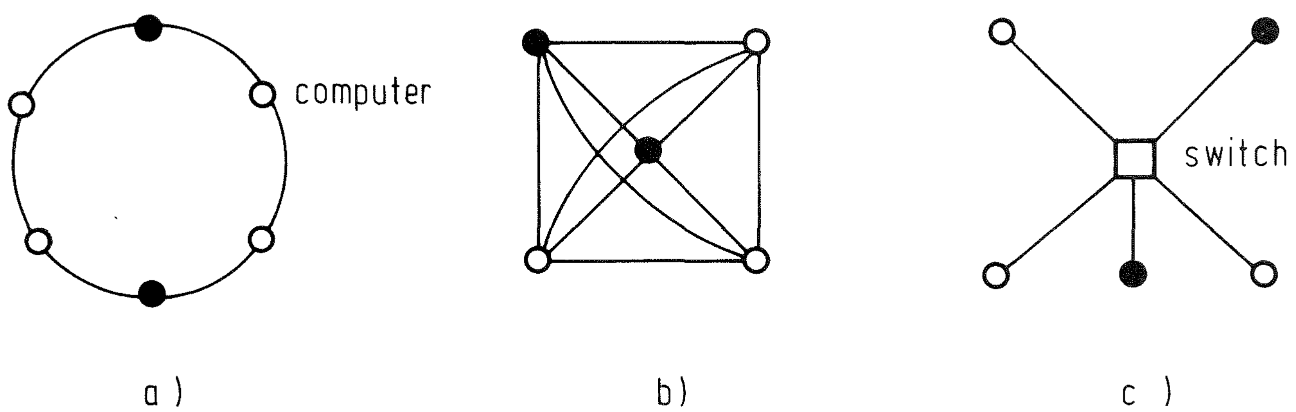


Fig. 2 Examples of symmetrical nets

filled circles indicate optimal allocation
of copies of a file

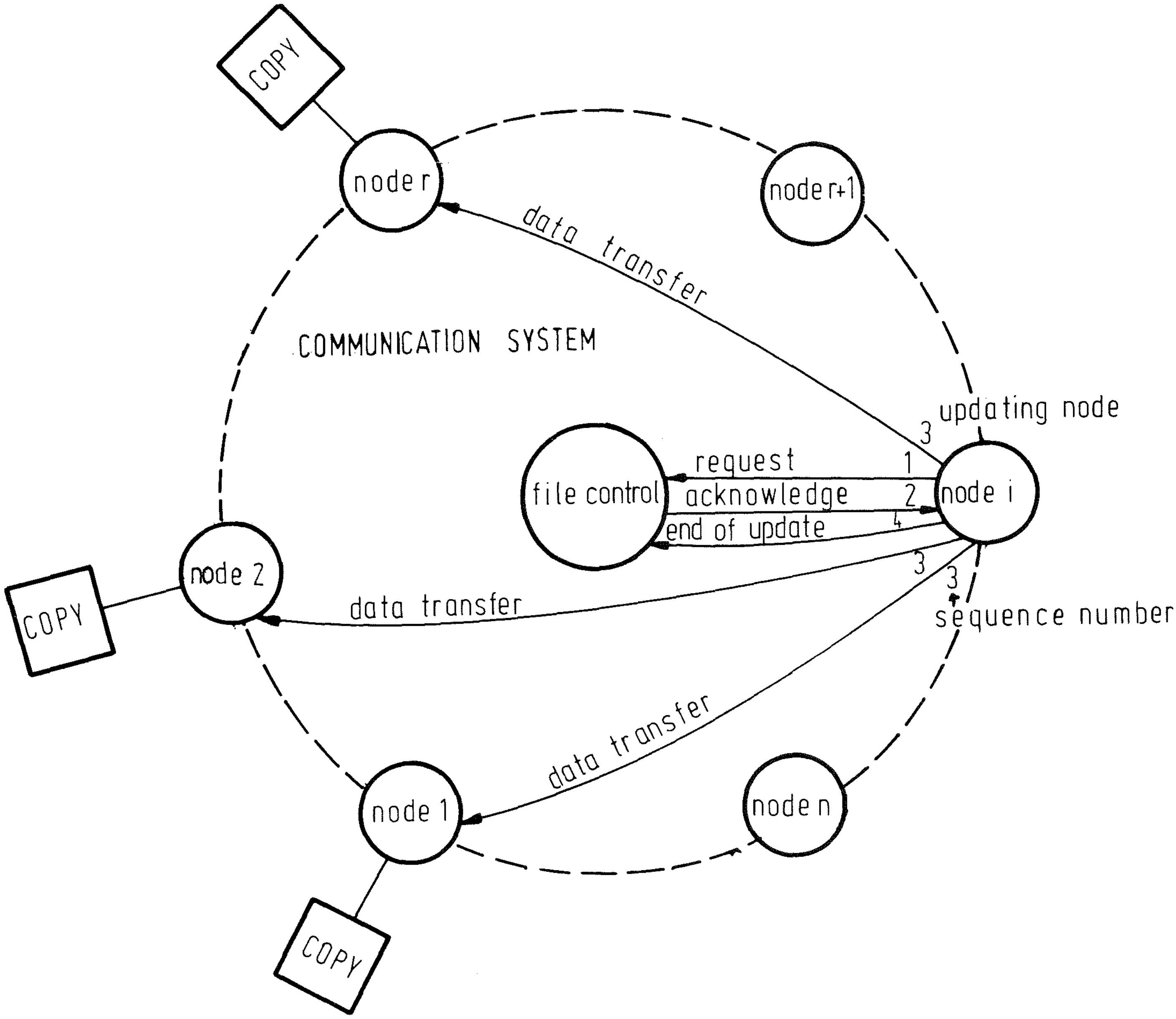
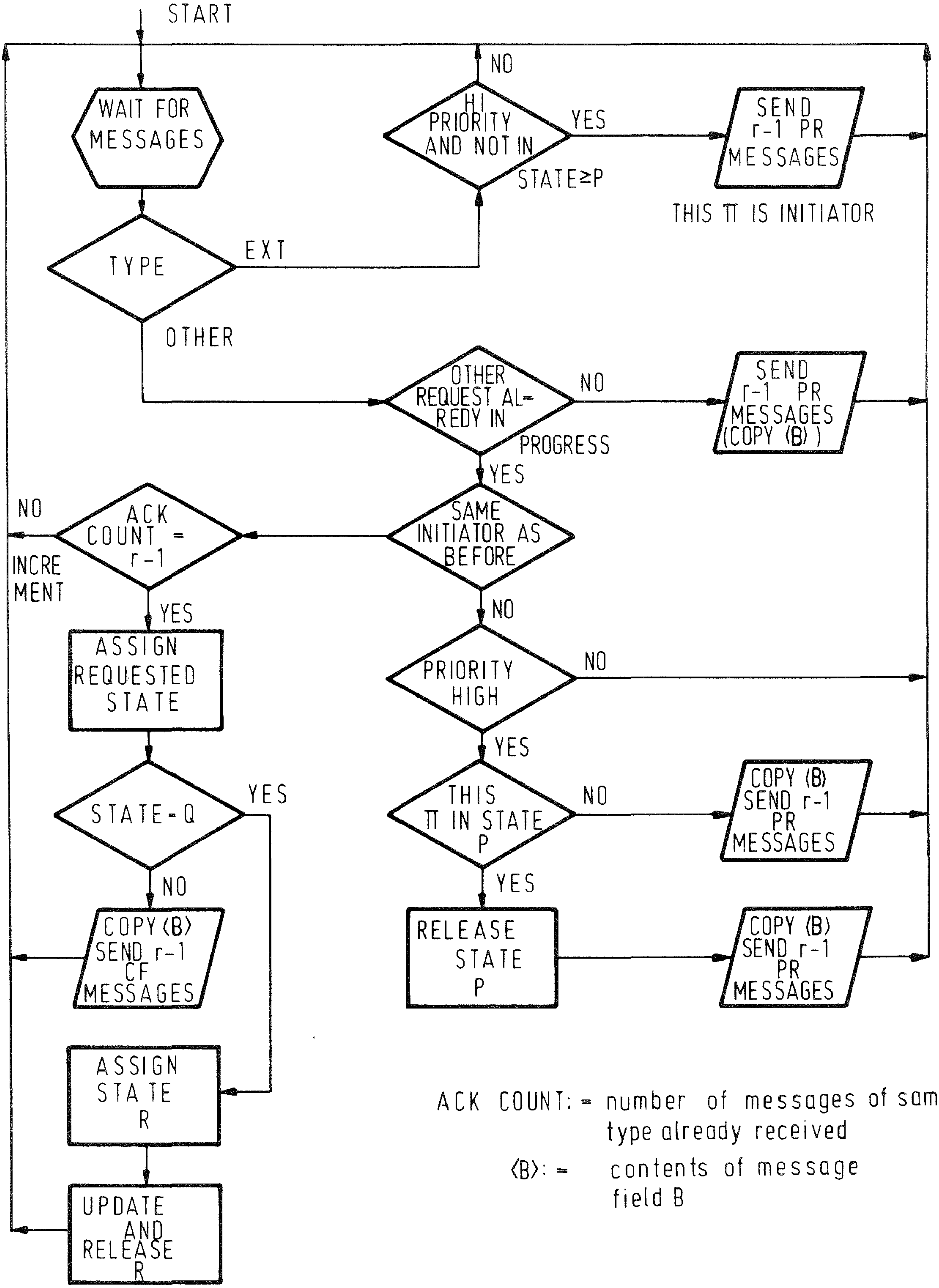


Fig. 3 Centralized synchronization of updating a file with copies located at nodes 1, ..., r



ACK COUNT: = number of messages of same type already received

⟨B⟩: = contents of message field B

Fig. 4 Production and consumption of consumable resources (messages) by a file manager

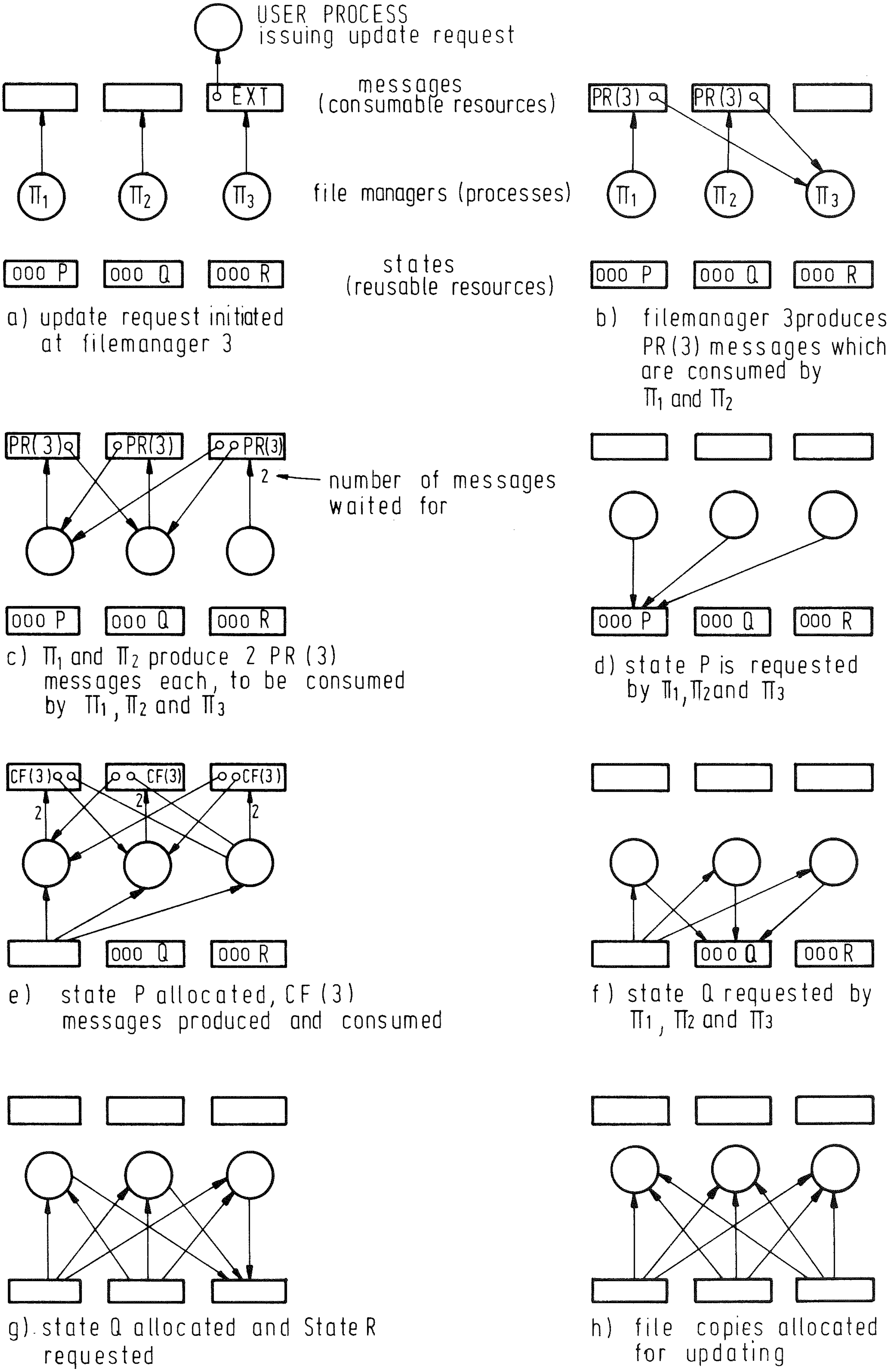


Fig. 5 States of the file system shown as general resource graphs