

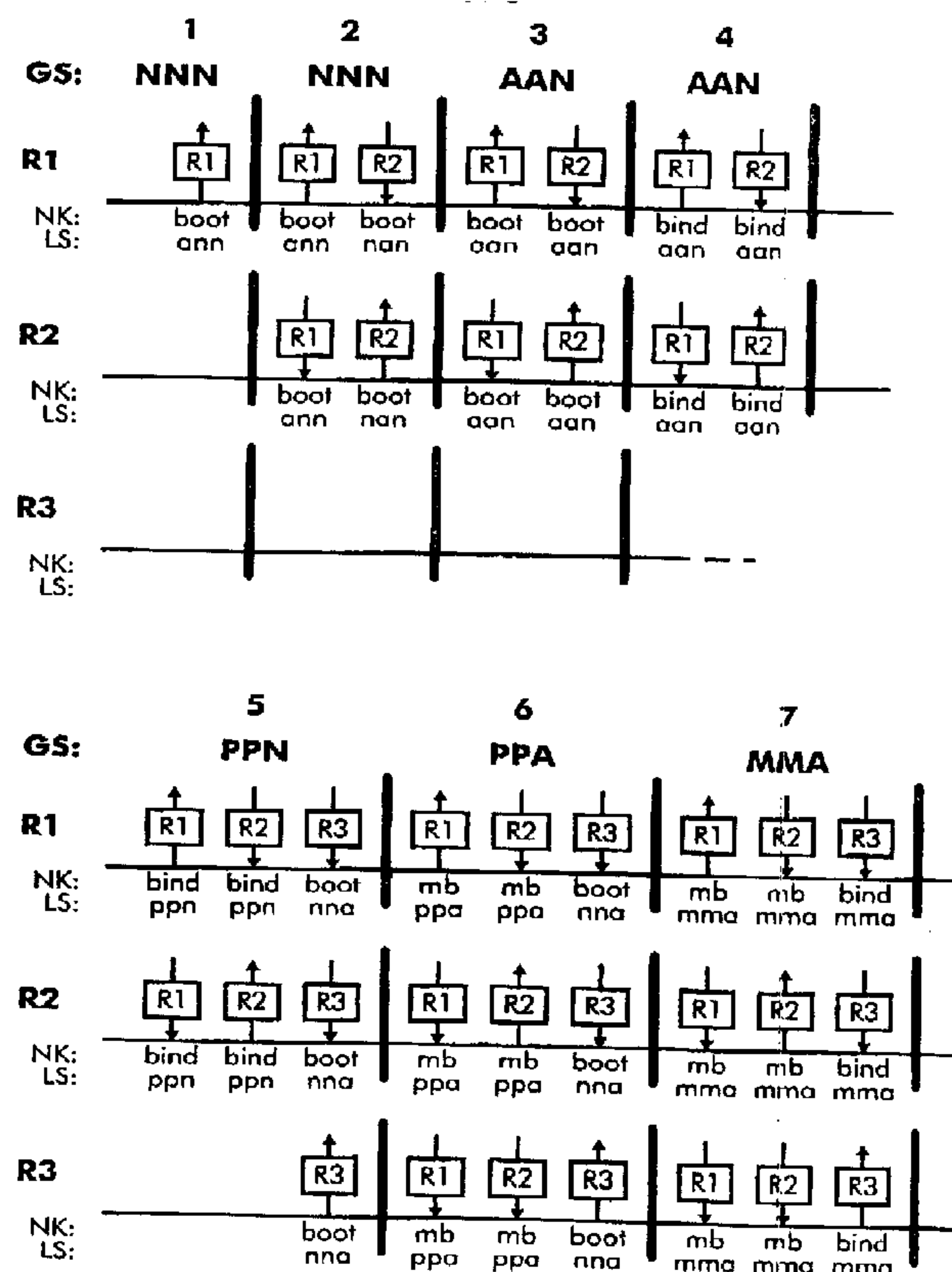
(72) KANTZ, HEINZ, AT  
(72) SCHECK, OLIVER, AT  
(72) METZNER, PETER, AT  
(71) ALCATEL, FR

(51) Int.Cl.<sup>6</sup> H04L 12/26

(30) 1998/07/15 (198 31 720.4) DE

(54) **METHODE POUR AVOIR UNE VUE GLOBALE UNIFORME DE  
L'ETAT D'UN SYSTEME DANS UN RESEAU  
INFORMATIQUE REPARTI**

(54) **METHOD OF DETERMINING A UNIFORM GLOBAL VIEW OF  
THE SYSTEM STATUS OF A DISTRIBUTED COMPUTER  
NETWORK**



(57) A method of determining a uniform global view (GS) of the system status of a distributed computer network (VRNW) comprising at least three computers (R1, R2, R3) is provided which is characterized in (a) that the computers exchange messages over communication links (KV12, KV13, KV32) in successive transmission rounds, with each of the computers receiving a message from each of the other computers in each transmission round in the absence of an error, (b) that in each transmission round, each of the computers evaluates messages received from the other computers and, based on the result of the evaluation, assigns one of at least three differently defined computer states to each of the other computers and determines a local view (LS) of the system status therefrom, and (c) that the computers exchange the local views of the system status determined by them, and each of the computers determines a uniform global view (GS) of the system status from the local views exchanged.



## Abstract of the Disclosure

A method of determining a uniform global view (GS) of the system status of a distributed computer network (VRNW) comprising at least three computers (R1, R2, R3) is provided which is characterized in

- (a) that the computers exchange messages over communication links (KV12, KV13, KV32) in successive transmission rounds, with each of the computers receiving a message from each of the other computers in each transmission round in the absence of an error,
- (b) that in each transmission round, each of the computers evaluates messages received from the other computers and, based on the result of the evaluation, assigns one of at least three differently defined computer states to each of the other computers and determines a local view (LS) of the system status therefrom, and
- (c) that the computers exchange the local views of the system status determined by them, and each of the computers determines a uniform global view (GS) of the system status from the local views exchanged.

Method of Determining a Uniform Global  
View of the System Status of a  
Distributed Computer Network

This invention relates to a method of determining a uniform global view of the system status of a distributed computer network comprising at least three computers. The invention further relates to a distributed computer network for carrying out the method.

INTRODUCTION

In distributed computer networks, changes in system status occasionally occur as a result of intended events (e.g., addition of a new computer) or unintended events (e.g., failure of a computer). On the occurrence of such a change, it must be ensured that the computers in the computer network get a uniform global view of the new system status as quickly as possible. The problem of how to bring about a uniform global view of the system status is frequently also referred to as a "membership problem".

This membership problem is particularly important in distributed computer networks which are used to monitor and control processes critical with regard to safety, such as in railway signaling or in power plant

technology. In such computer networks, the individual computers compare their results. Results are output to the process only if they were determined independently of each other by a majority of the computers. If, in a network of three computers, for example, one of the computers fails, the other two computers can continue to deliver results to the process. This requires, however, that these two computers have come to a uniform global view of the system status, i.e., there must be agreement upon which of the computers has failed and which of the computers are free from faults.

From a publication by L. E. Moser et al entitled "Membership Algorithms for Asynchronous Distributed Systems", 11th Int. Conf. on Distributed Computing Systems, Arlington, TX, USA, May 1991, pages 480-488, different algorithms for solving the membership problem in an uncoupled distributed computer network are known. These algorithms are based on a failure hypothesis according to which the computers send either no messages or correct messages. The case where a computer sends erroneous messages is not assumed. The algorithms described use messages whose transmission is repeated if a receiver has not received the message. In addition, there are messages whose transmission is not repeated in such a case. This latter group includes, for example, the request messages, by which a computer notifies the other computers that it wants to become a member again. Admission to such a request is granted by the other computers via specific grant messages. The algorithms described there are limited to uncoupled computer networks and cannot readily be applied to synchronous or virtually synchronous distributed networks.

## OBJECT

It is therefore an object of the invention to provide a method of determining a uniform global view of the system status of a synchronous or virtually synchronous distributed computer network comprising at least three computers. Another object of the invention is to provide a distributed computer network for carrying out the method.

## SUMMARY OF THE INVENTION

These objects are attained by the features recited in claims 1 and 7, respectively. According to the invention, communication among the computers is implemented in the form of transmission rounds. A transmission round is characterized in that in such a round, each of the computers receives a message from each of the other computers in the absence of an error. Each of the computers evaluates the messages received from the other computers and, based on the result of the evaluation, assigns one of at least three differently defined computer states to each of the other computers. In this manner, each computer determines its own local view of the system status. The computers exchange these local views. Each computer then determines a global view of the system status from the received local views, for example by subjecting the local views to a majority decision. As all of the computers have the same local views, they all come to the same global view of the system status.

This method places no exacting requirements on the synchrony of the transmission. It only requires that within a period of time which need not be fixed but

must be finite and limited, each of the computers has received a message from each of the other computers. The method can thus be used with communications protocols according to which computers may send only during permanently assigned time slots, but also with a few communications protocols where such a fixed assignment does not exist.

Furthermore, use of the method according to the invention requires no specific sequences of operations to start up the distributed computer network, whereby the complexity of the computer network is reduced significantly.

Further advantageous features of the invention are apparent from the subclaims.

The invention will become more apparent from the following description of embodiments to be read in conjunction with the accompanying drawings, in which:

Fig. 1 shows a distributed computer network with three computers R1, R2, and R3;

Fig. 2 is a sketch serving to explain the term "transmission round";

Fig. 3a is a sketch serving to explain the term "transmission round" if a strictly synchronous communications protocol is used;

Fig. 3b is a sketch serving to explain the term "transmission round" if a virtually synchronous communications protocol is used;

- Fig. 4 shows a distributed computer network with a faulty computer to explain the method in accordance with the invention;
- Fig. 5 shows the procedure according to the invention for transmission rounds in a distributed computer network as shown in Fig. 4;
- Fig. 6 shows the structure of a message according to the invention as claimed in claim 2;
- Fig. 7 shows a distributed computer network to explain an embodiment as claimed in claim 5;
- Fig. 8 shows a distributed computer network in which a Byzantine fault has occurred; and
- Fig. 9 shows the procedure according to the invention for transmission rounds during the booting of a distributed computer network in which four computer states are defined.

Fig. 1 shows a distributed computer network VRNW comprising three computers R1, R2, and R3. The computers R1, R2, and R3 are interconnected by communication links KV12, KV13, and KV32 in such a way that each of the computers can communicate with each of the other computers in the computer network. In Fig. 1, these communication links are represented as physical point-to-point connections. In many cases, however, it will be more advantageous to have the computers R1, R2, and R3 communicate over a data bus.

The communications protocol governing the communication between the computers R1, ..., R3 is packet-oriented and makes it possible to selectively address message packets - hereinafter called "messages" for short - to individual computers. The addressee is in a position to identify the sender with the aid of the message. In addition, the communications protocol is so designed that messages are exchanged in the form of transmission rounds. A transmission round is characterized in that in such a round, each of the computers receives a message from each of the other computers in the absence of an error. The term "transmission round" will now be explained in more detail with reference to Fig. 2 as well as to Figs. 3a and 3b.

Fig. 2 shows schematically the procedure for a transmission round in the distributed computer network VRNW of Fig. 1. It is assumed that computer R1 sends messages A and A' to the two other computers R2 and R3, respectively. The messages A and A' may be different or identical. Computers R2 and R3 respond to the reception of the messages A and A', respectively, by transmitting messages B, B' and C, C', respectively. This results in a state in which each of the computers has received a message from each of the other computers.

How the exchange of the messages is implemented depends on the communications protocol used. Fig. 3a shows schematically the procedure for transmission rounds in a tightly coupled synchronous distributed computer network. The time axis is divided into a plurality of cycles  $Z_i$ , whose limits are marked in Fig. 3a by grey bars. In each cycle, each computer is assigned a time slot during which it may send messages over the

communication links. For the timing of the time slots assigned to them, the computers must have a common time base.

In cycle  $Z_1$ , the first time slot is assigned to computer R1. It therefore begins to send messages to the other two computers R2 and R3. Due to possibly different propagation delays, the messages are received by the other computers R2 and R3 not simultaneously, but still within the time slot assigned to computer R1. The next time slot in cycle  $Z_1$  is assigned to computer R2, which now sends messages to the other computers, i.e., to computer R1 and computer R3. Computer R3 proceeds analogously. At the end of cycle  $Z_1$ , each of the computers has received a message from each of the other computers. The subsequent cycles  $Z_{1+1}$  and  $Z_{1+2}$  follow the same pattern.

The procedure of transmission rounds depicted in Fig. 3b is based on a virtually synchronous, loosely coupled distributed computer network, which does not require a common time base. In the embodiment shown, the first transmission round  $i$  is initiated by computer R1 sending messages to the two other computers R2 and R3. As soon as the two other computers R2 and R3 have received the messages, they, in turn, send messages to the other computers R1, R3 and R1, R2, respectively. Here, however, no time slots within which the computers may send messages and within which the messages must be received are fixed. Coupling between the computers thus exists only insofar as the computers R2 and R3 respond to the reception of the message from computer R1 with the transmission of messages of their own.

In this virtually synchronous computer network, too, a transmission round is complete when each of the computers has received a message from each of the other computers. This point of time is generally different for each computer; this is indicated by the curved vertical bar. If, for example, computer R3 received no message from computer R2, the transmission round would not be complete for computer R3 until after expiration of a defined waiting time.

In this virtually synchronous distributed computer network, unlike the synchronous computer network described above, any of the computers can initiate a transmission round. The next transmission round  $i + 1$  can therefore be initiated by computer R2, for example, as shown in Fig. 2b. In this transmission round  $i + 1$ , computer R1 is the first to detect the end of the round. Hence, it can already initiate the next transmission round  $n + 2$  although transmission round  $n + 1$  has not yet been completed for the other two computers.

### THREE COMPUTER STATES

According to the invention, in each transmission round, each of the computers evaluates messages received from the other computers and, based on the result of the evaluation, assigns one of at least three differently defined computer states to each of the other computers. In a first embodiment of the invention, a received message can be assigned to one of the three following categories:

- (a) Member message: Is sent by a computer which is a member. A member responds reliably in each transmission round with the transmission of messages of its own when it has received a message.
- (b) Applicant message: Is sent by a computer which is not yet a member but wants to become one.
- (c) Null message: Either no message has been received or the receiving computer has determined by evaluation of an error-detecting code that the received message is corrupted.

A computer which has received a message from another computer will now assign one of the three following computer states to the sending computer depending on which of these categories the message belongs to:

- (a) "Member" (abbreviation: m). A computer which has sent a member message.
- (b) "Applicant" (abbreviation: a). A computer which has sent an applicant message.
- (c) "Nonmember" (abbreviation: n). A computer which has sent a null message.

In the example sketched in Fig. 4, computer R1 has started a transmission round  $i$ , which is indicated by the dashed arrows. Computer R3 is faulty; therefore, it sends a message neither to computer R1 nor to computer R2. The computers R1 and R2 interpret this as "null messages" and, consequently, assign the state

"nonmember" (n) to the faulty computer R3. As the computers R1 and R2 have successfully exchanged member messages A and B, they classify each other as "members" (m). In addition, computers R1 and R2 consider themselves to be members.

These computer states are composed by each of the computers into a local view of the distributed computer network. Thus, in the local view determined by computer R1 in transmission round  $i$  (abbreviated:  $LS_{i,1}$ ), computers R1 and R2 are members (m), while computer R3 is a nonmember (n). This is symbolized in Fig. 4 by the expression  $LS_{i,1} = mmn$ . The local view determined by computer R2 in transmission round  $i$  (abbreviated:  $LS_{i,2}$ ) is also  $mmn$ . Computer R3 has received a member message from all other computers and could regard itself and the other computers as members. However, computer R3 could just as well be so seriously faulty as to be unable to determine a local view of the system status at all. For this reason, the local view determined by computer R3 is designated by the expression "???".

In the next transmission round  $i + 1$ , according to the invention, the three computers  $R_x$ ,  $x = 1, 2, \text{ or } 3$ , exchange the local views  $LS_{i,x}$  determined by them in round  $i$ . Fig. 5 shows this in a schematic representation. In transmission round  $i$ , only computers R1 and R2, but not the faulty computer R2, send messages. In transmission round  $i + 1$ , computers R1 and R2 send messages which inform a receiver of the respective local view determined by the sender for the preceding transmission round  $i$ . Thus, at the end of transmission round  $i + 1$ , each of the two computers R1 and R2 has its own view and the local view determined

by the respective other computer. Since computer R3 does not send a message in transmission round  $i + 1$ , either, it cannot notify the other computers of its local view of the system status, provided such a local view has been determined.

Through the exchange of local views, the method according to the invention ensures that at the end of transmission round  $i + 1$ , at least each correctly functioning computer holds the same local views. If all computers determine a global view of the system status from the local views according to the same rule, it is reliably ensured that these global views are identical and, thus, uniform. Therefore, the following speaks only of one global view  $GS_i$  for transmission round  $i$ .

One possible rule for determining a global view from the individual local views is to subject the local views to a simple majority decision. This means that in the global view, a computer is assigned that computer state in which the computer is in the majority of the local views. In the above example with only two local views, which are identical, the result is, of course:

$$\begin{array}{rcl}
 LS_{i,1}: & m & m & n \\
 LS_{i,2}: & m & m & n \\
 \hline
 GS_i: & M & M & N
 \end{array}$$

where capital letters denote a computer state in a global view. If the faulty computer R3 sent its local view to the two other computers R1 and R2 in transmission round  $i + 1$ , this would not alter the

global view of these computers, since they would still be in the majority:

$LS_{i,1}:$	m	m	n
$LS_{i,2}:$	m	m	n
$LS_{i,3}:$	?	?	?
$GS_i:$	M	M	N

This also shows that the global view of the system status in transmission round  $i$  cannot be determined before the next transmission round  $i + 1$  is complete, i.e., before the local views of round  $i$  have been exchanged.

The simple majority decision just explained can be replaced by a more qualified majority decision. For example, in a distributed computer network with five computers, it could be agreed that in the global view, a computer is regarded as a member (M) only if it is a member (m) in not only three local views, but in at least four local views. On the other hand, under certain circumstances, less than a simple majority decision may be required; reference is made to the below-explained embodiment in which four computer states are defined.

In the determination of the global views, not only the local views exchanged, but also the global view of a preceding transmission round may be taken into account. It may be agreed, for example, that only the local views of those computers which were members in the last determined global view are taken into account. Likewise it may be agreed that it depends on the last determined

global view what category of messages a computer may send. An example of this will be explained later.

The procedure according to the invention preferably extends to all transmission rounds, i.e., in each transmission round, local views are exchanged between the computers. Depending on the respective application, however, it is also conceivable to perform the determination of local views and their subsequent exchange only in response to a particular initiation signal or at (major) fixed intervals (e.g., every tenth transmission round). The exchange of local views is independent of the exchange of other messages.

In a preferred embodiment, the messages exchanged between the computers all have the format shown in Fig. 6. Each message comprises a header HEAD which contains the following information:

- (a) message category NK;
- (b) number  $i$  of the transmission round; and
- (c) local view  $LS_{i,j}$  of the system status, determined by computer  $j$  in transmission round  $i$ .

If required by the application being executed by the distributed computer network, the messages may include a payload ND. If - as shown in Fig. 4, for example - a computer sends messages A and A' to all other computers in a transmission round, these messages A and A' may only differ in their payloads ND. The header HEAD, however, must be identical in both messages, since

otherwise no uniform view of the system status can be determined.

#### FAULT TOLERANCE

If, as shown in Fig. 4, each of the computers receives precisely one message from each of the other computers in a transmission round, the above-described method according to the invention can be used to reliably determine a uniform global view of the system status if the fault in a computer can be assigned to one of the two following error classes:

1. The computer is sending no messages at all.
2. The computer is sending erroneous messages, with the error manifesting itself to all other computers in the same manner, however.

The method according to the invention can be used to advantage with a communications protocol which ensures that a data packet sent by any of the computers to the other computers is forwarded by these other computers in such a way that in the absence of an error, each of the other computers will receive this data packet at least twice. This will now be explained in more detail with reference to Fig. 7. In the distributed computer network shown in Fig. 7, computer R1 sends identical messages A to computers R2 and R3. Computer R2 receives the message A and forwards it to computer R3 unchanged. Computer R3 handles the message received from computer R1 in the same way. In this manner, each of computers R2 and R3 receives the message A twice, namely directly from computer R1 and indirectly via computers R3 and R2, respectively.

With such a communications protocol, the method according to the invention can be used to determine a uniform global view of the system status even if the fault in a computer can be assigned to a third error class. In this error class, as in the above-mentioned 2nd error class, the faulty computer sends erroneous messages. However, the fault may manifest itself to the other computers in different ways (so-called Byzantine fault).

Such a fault scenario is sketched in Fig. 8. Computer R1 is faulty and generates different messages A and A' instead of identical messages. Both messages are provided with "correct" error-detecting codes, so that the two other computers R2 and R3 cannot detect any error. If, however, the communications protocol ensures in the above-mentioned manner that the messages are forwarded, then computer R2, for example, will determine that message A, received directly from computer R1, is not identical with message A', received via computer R3. Computer R3 will make a corresponding determination. Because of this discrepancy, in the local views determined by the two correctly functioning computers R2 and R3, computer R1 will be regarded as a nonmember. It is only assumed that no errors occur during the forwarding of messages; otherwise it could no longer be distinguished for computer R2, for example, whether computer R1 has sent different messages or whether computer R3 has corrupted a message A received from computer R1 into a message A'.

If such a communications protocol is used, a uniform global view of the system status can be determined even

if a communication link between the computers is faulty. If, for example, the link between computers R1 and R2 in Fig. 7 fails, the two affected computers R1 and R2 can continue to exchange messages via computer R3. If the path the messages have taken is additionally specified in the messages, then, as is readily apparent, all computers can determine which of the communication links is faulty.

#### FOUR COMPUTER STATES

In a preferred embodiment, the computers are assigned four rather than three differently defined states. Accordingly, there are four differently defined message categories:

- (a) Member message (abbreviation: mb): Is sent by a computer which is a member. A member responds reliably in each transmission round with the transmission of own messages when it has received a message.
- (b) Bind message (abbreviation: bind): Is sent by a computer which is not yet a member but is already regarded as an applicant (see below) in the global view.
- (c) Boot message (abbreviation: boot): Is sent by a computer which is in a booting process or has completed such a process.
- (d) Null message (abbreviation: null): Either no message has been received or the receiving computer has determined by evaluating an error-detecting code that the received message is corrupted.

The following four computer states correspond to these four different message categories:

- (a) "Member" (abbreviation: m). A computer which has sent a member message.
- (b) "Preliminary member" (abbreviation: p): A member which has sent a bind message.
- (c) "Applicant" (abbreviation: a): A computer which has sent a boot message.
- (d) "Nonmember" (abbreviation: n): A computer which has sent a null message.

The meanings of these message categories and of the computer states as well as the advantages resulting from these definitions will now be illustrated by an example.

#### *Booting of a Distributed Computer Network*

Fig. 9 shows schematically how a uniform global view of the system status can be determined with four computer states even while booting the distributed computer network. In this example it is assumed that the computer R1 boots first, followed directly by computer R2, and some time later by computer R3. Consequently, the first transmission round 1 begins with a boot message (boot) sent by computer R1 to the other computers. In Fig. 9, the respective message categories NK as well as the respective local views LS determined by the computers are given. Also given are the global views determined by the computers.

As computer R1 cannot have received any messages from other computers, it assigns the state "nonmember" (n) to these computers. It regards itself as an applicant (a), since it sends a boot message. As no other local view is available yet, a majority decision automatically yields the lowest possible global system status, namely  $GS_1 = NNN$ .

In transmission round 2, computer R1 sends a boot message in addition to computer R2. Therefore, computers R1 and R2 assign to each other the state "applicant" (a), so that the local views are  $LS_{2,1} = LS_{2,2} = aan$ . From the preceding transmission round, only one local view is available, so that a majority decision on the local views cannot produce a result. Hence, the global system status remains NNN.

In transmission round 3, computers R1 and R2 send boot messages again and exchange their local views determined in the preceding transmission round 2. As the two local views are identical, both computers determine the global view as  $GS_3 = AAN$ .

Since, in the global view, both computer R1 and computer R2 are applicants, they can send bind messages (bind) in the next transmission round, i.e., round 4. Upon reception of bind messages, computers R1 and R2 assign to each other the computer state "preliminary member" (p), so that the local views exchanged by these two computers in transmission round 5 are  $LS_{4,1} = LS_{4,2} = ppn$ .

In transmission round 5, the computers still send bind messages, since in the previously determined global view, they were still applicants (A). The identical local views  $LS_{4,1} = LS_{4,2} = \text{ppn}$  result in the global view  $GS_5 = \text{PPN}$ . In transmission round 5, computer R3 sends a boot message for the first time, so that at the end of the next transmission round, round 6, it is already regarded as an applicant (A) in the global view. In this transmission round, both computers R1 and R2 can send member messages for the first time, since in the preceding transmission round, they were considered preliminary members in the global view. In transmission round 7, both computers R1 and R2 are members in the global view. The distributed computer network now has a majority of members.

This example illustrates how a hierarchy of computer states in which the computers can rise step by step from transmission round to transmission round is created by the method according to the invention. In the example just explained, a computer will rise in the global view only if a majority of the local views are in agreement. It may be advantageous, however, if a computer rises in the global view already when it is seen at the higher hierarchy level by a single computer which is considered a member in the global view. A majority vote is still required only for the rise to the globally highest hierarchy level (M). In this manner, the rise in the hierarchy can in many cases be speeded up.

If a computer is found to be faulty, it is preferably agreed that in the local views determined by the computers, it immediately descends to the lowest

computer state, i.e., nonmember (n). A stepwise descent is also possible. To which hierarchy level the computer has to descend can be made dependent, for example, on the type of fault that has occurred, if this can be determined.

(Re)Admission of an individual computer to the distributed computer network is analogous to the sequence just described. How the last booting computer R3 is admitted to the computer network can be derived with the aid of Fig. 9. In transmission round 8 (not shown), computer R3 sends bind messages (bind) again. At the end of this round, it is already regarded as a preliminary member (p) in the global view. In transmission round 9, computer 3 sends member messages (mb), and in the next round, round 10, it is considered a member (m) in the global view.

It is to be understood that the use of the method according to the invention is not limited to distributed computer networks with three computers. It is a special advantage of the method that it is extremely flexible in terms of the number and type of computers, the underlying communications protocol, and the possible applications of the computer system. In particular, neither special hardware nor a specific operating system is required for the computers.

## Patent Claims

1. A method of determining a uniform global view (GS) of the system status of a distributed computer network (VRNW) comprising at least three computers (R1, R2, R3),

c h a r a c t e r i z e d i n

- (a) that the computers exchange messages over communication links (KV12, KV13, KV32) in successive transmission rounds, with each of the computers receiving a message from each of the other computers in each transmission round in the absence of an error,
- (b) that in each transmission round, each of the computers evaluates messages received from the other computers and, based on the result of the evaluation, assigns one of at least three differently defined computer states to each of the other computers and determines a local view ( $LS_{i,j}$ ) of the system status therefrom, and

- (c) that the computers exchange the local views of the system status determined by them, and each of the computers determines a uniform global view of the system status from the local views exchanged.
2. A method as claimed in claim 1 wherein the messages contain at least the following information:
    - (a) a message category (NK);
    - (b) the number (i) of the transmission round in which the message is sent;
    - (c) a local view ( $LS_{i,j}$ ) of the system status, determined by the sending computer.
  3. A method as claimed in either of the preceding claims wherein the determination of the global view of the system status also takes into account the global view of a preceding transmission round.
  4. A method as claimed in any one of the preceding claims wherein the determination of the global view of the system status involves a majority decision on the individual local views.
  5. A method as claimed in any one of the preceding claims wherein a message sent by any of the computers to the other computers is forwarded by said other computers in such a manner that in the absence of an error, each of the other computers will receive said message at least twice.

6. A method as claimed in any one of the preceding claims wherein each transmission round comprises the following steps:
- (a) Any of the computers (R1 in Fig. 2) sends, on its own initiative, a data packet (A, A') to each of the other computers (R2, R3) in the distributed computer network; and
  - (b) each of the other computers (R2, R3) responds to the reception of the data packet (A, A') within a predeterminable period of time with the transmission of a data packet (B, B'; C, C') of its own to all other computers in the computer network.
7. A distributed computer network (VRNW) comprising at least three computers (R1, R2, R3) and means for determining a uniform global view (GS) of the system status of the computer network,
- c h a r a c t e r i z e d   i n
- (a) that communication links (KV12, KV13, KV32) are provided over which the computers exchange messages in successive transmission rounds, with each computer receiving a message from each of the other computers in each transmission round in the absence of an error,
  - (b) that each of the computers has a processor which evaluates the messages received from the other computers in a transmission round and, based on the result of the evaluation, assigns one of at

least three differently defined computer states to each of the other computers and determines a local view ( $LS_{i,j}$ ) of the system status therefrom, and

- (c) that the respective processor causes the computers to exchange the local views of the system status determined by them, and determines a uniform global view of the system status from the local views exchanged.
8. A distributed computer network as claimed in claim 7 wherein the messages contain at least the following information:
- (a) a message category (NK);
  - (b) the number (i) of the transmission round in which the message is sent;
  - (c) a local view ( $LS_{i,j}$ ) of the system status, determined by the sending computer.
9. A distributed computer network as claimed in claim 7 or 8 wherein the global view of a preceding transmission round is taken into account by the processor in determining the global view of the system status.
10. A distributed computer network as claimed in any one of claims 7 to 9 wherein the processor determines the global view of the system status by a majority vote.

11. A distributed computer network as claimed in any one of claims 7 to 10 wherein a data packet sent by any of the computers to the other computers is forwarded by said other computers in such a manner that in the absence of an error, each of the other computers will receive said data packet at least twice.
  
12. A distributed computer network as claimed in any one of claims 7 to 11 wherein each transmission round comprises the following steps:
  - (a) Any of the computers (R1 in Fig. 2) sends, on its own initiative, a data packet (A, A') to each of the other computers (R2, R3) in the distributed computer network; and
  
  - (b) each of the other computers (R2, R3) responds to the reception of the data packet (A, A') within a predeterminable period of time with the transmission of an own data packet (B, B'; C, C') to all other computers in the computer network.

1/6

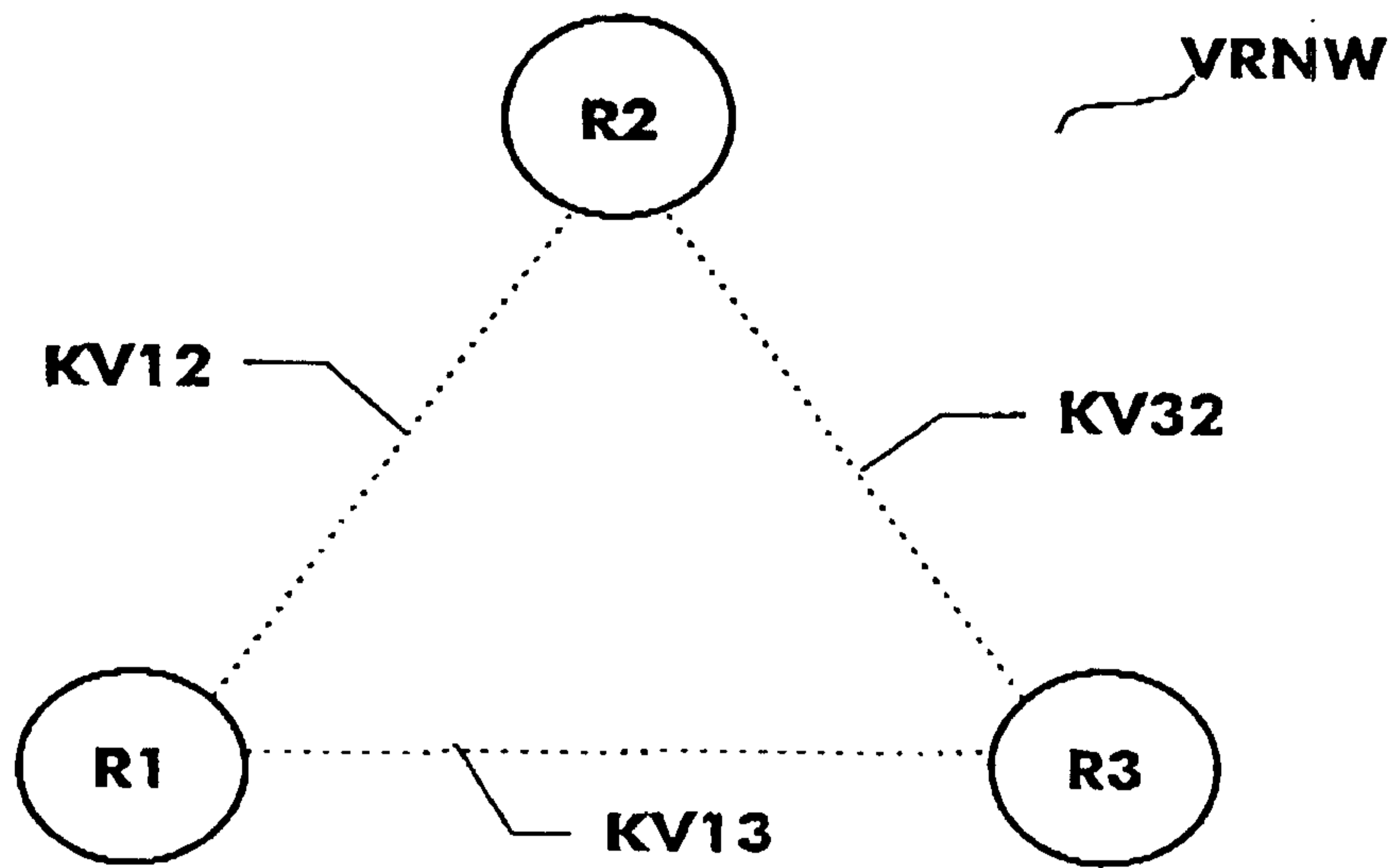


Fig.1

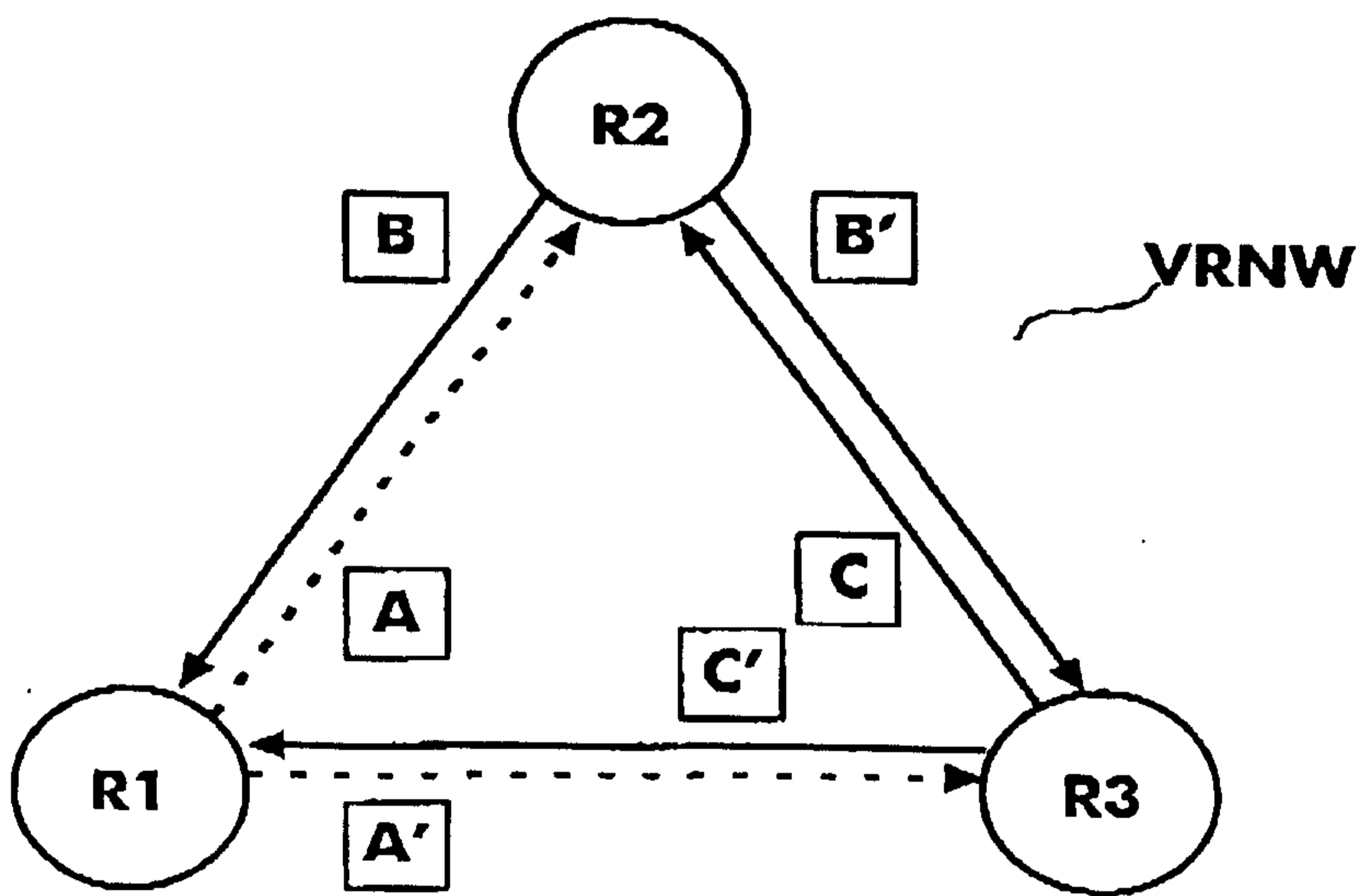


Fig.2

2/6

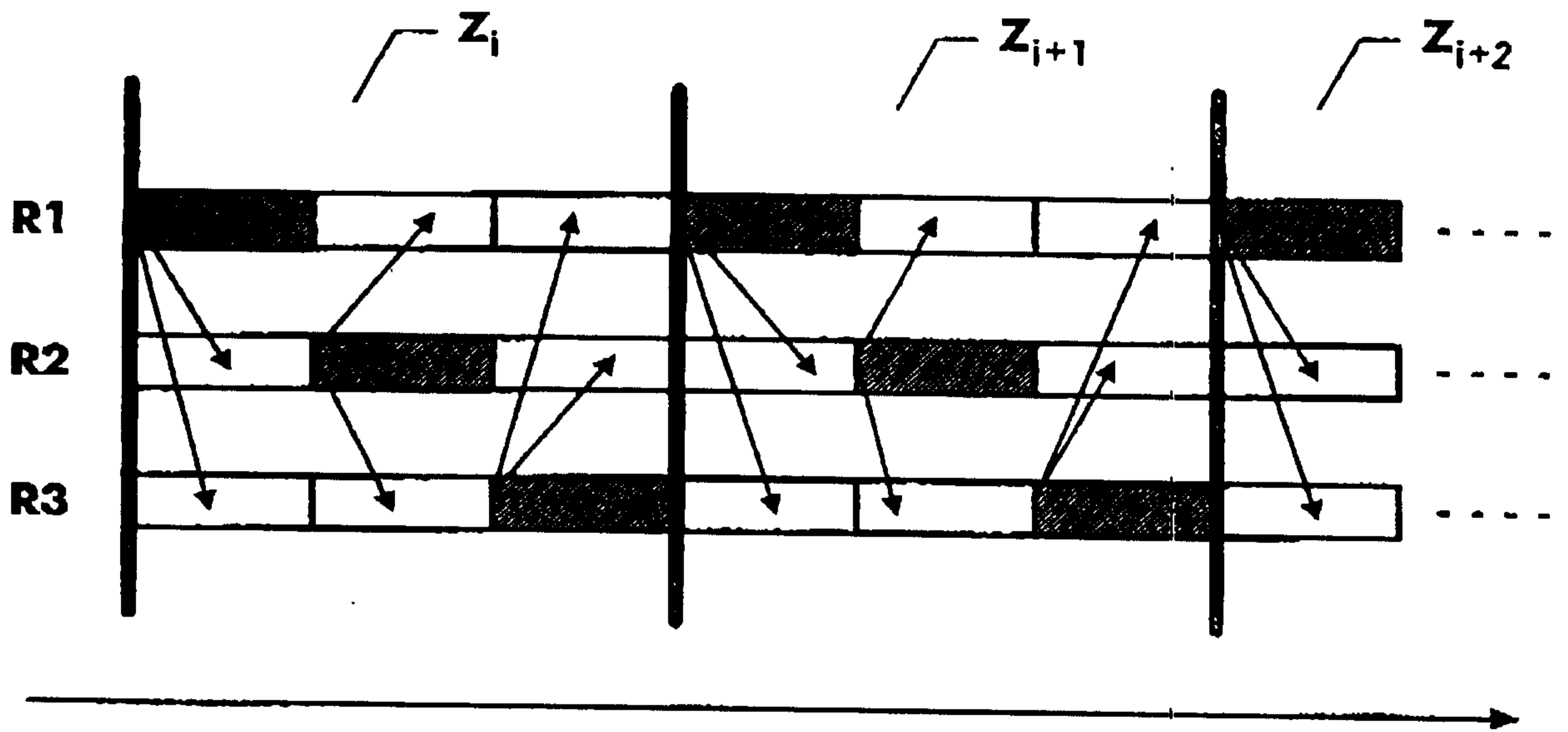


Fig.3a

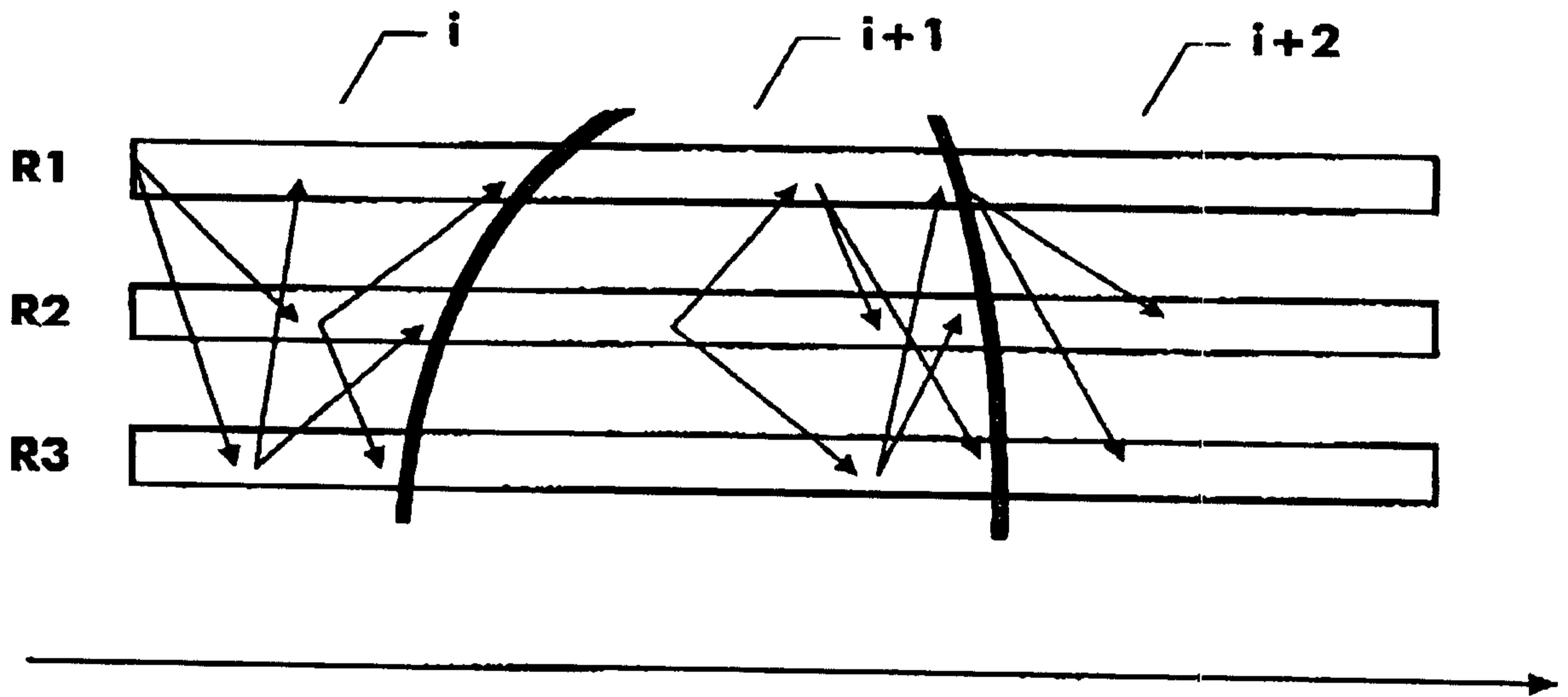


Fig.3b

3/6

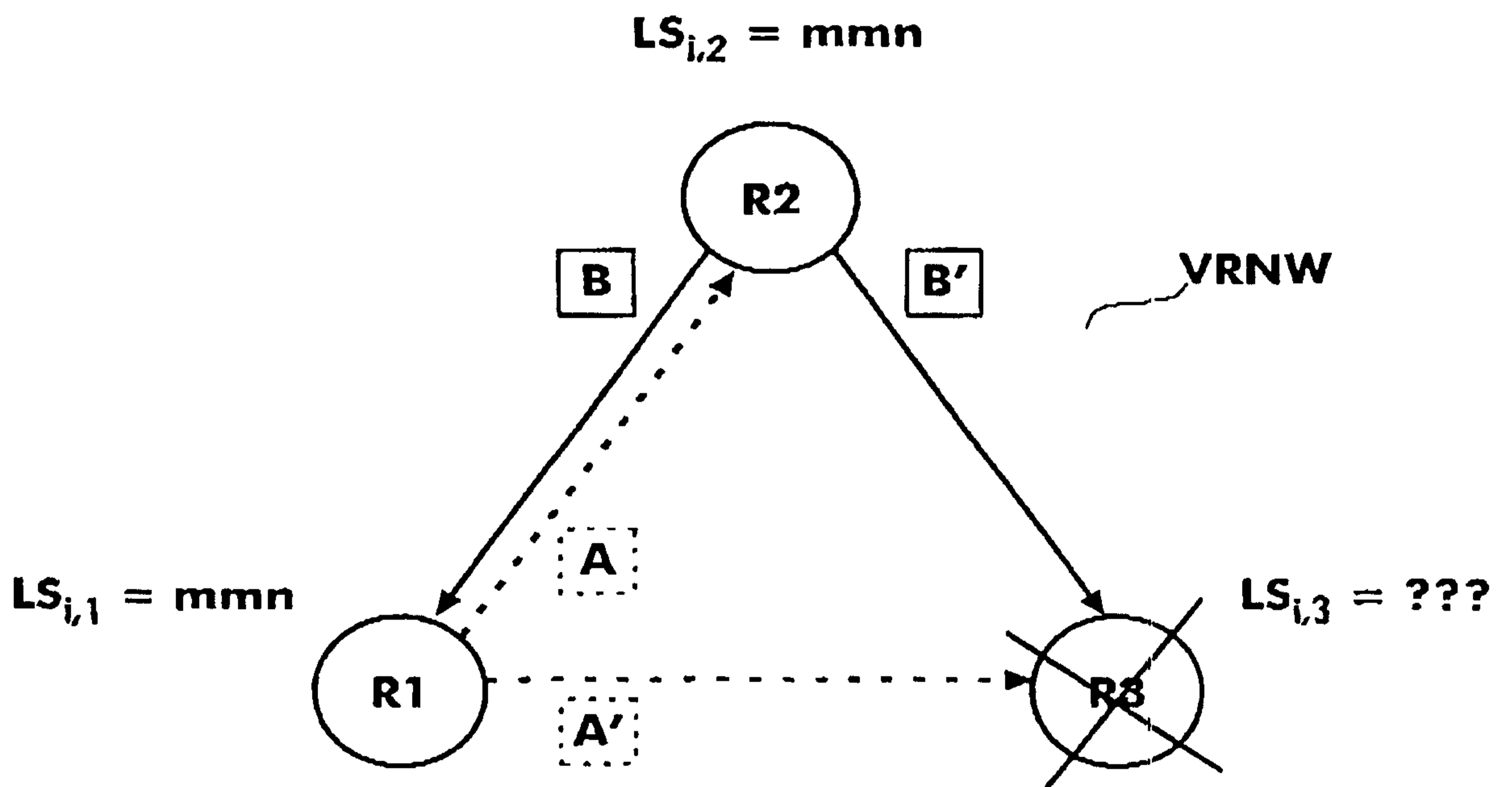


Fig.4

4/6

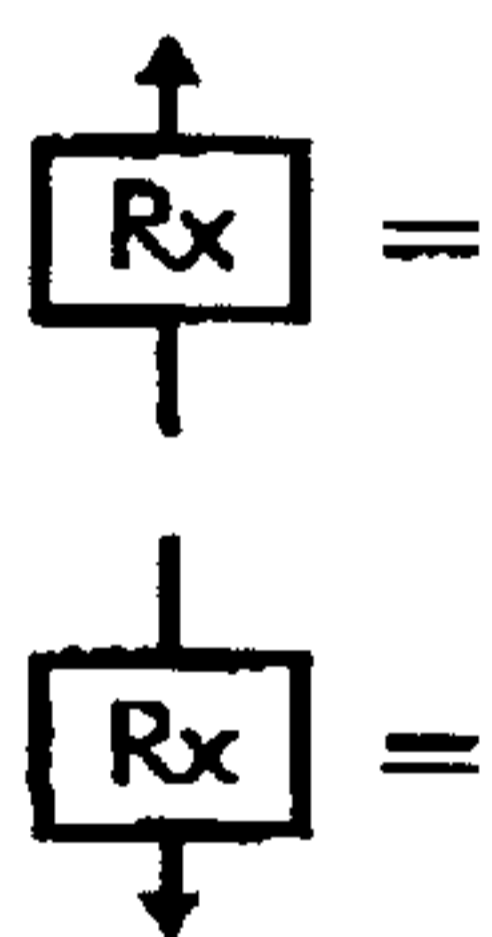
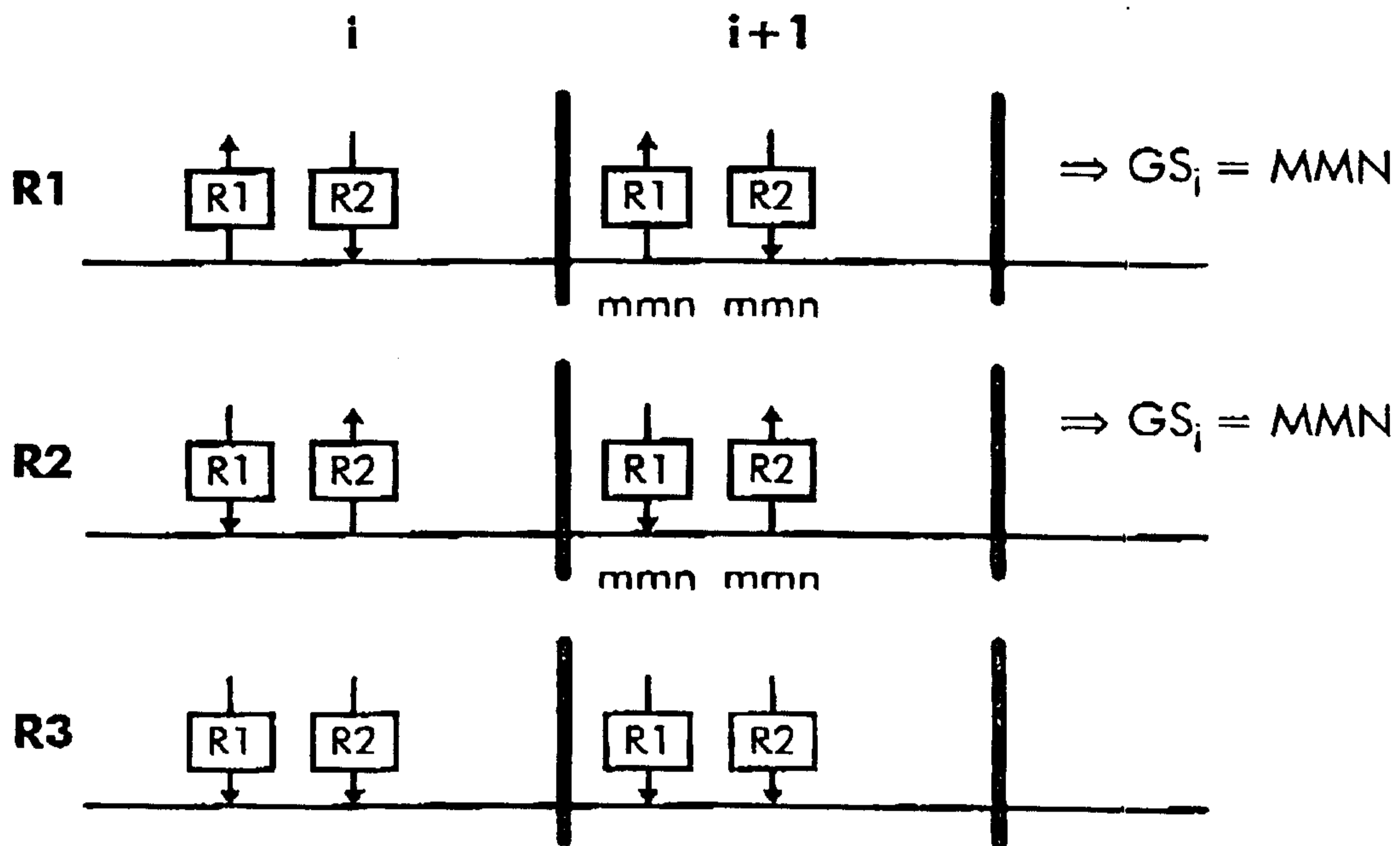


Fig.5

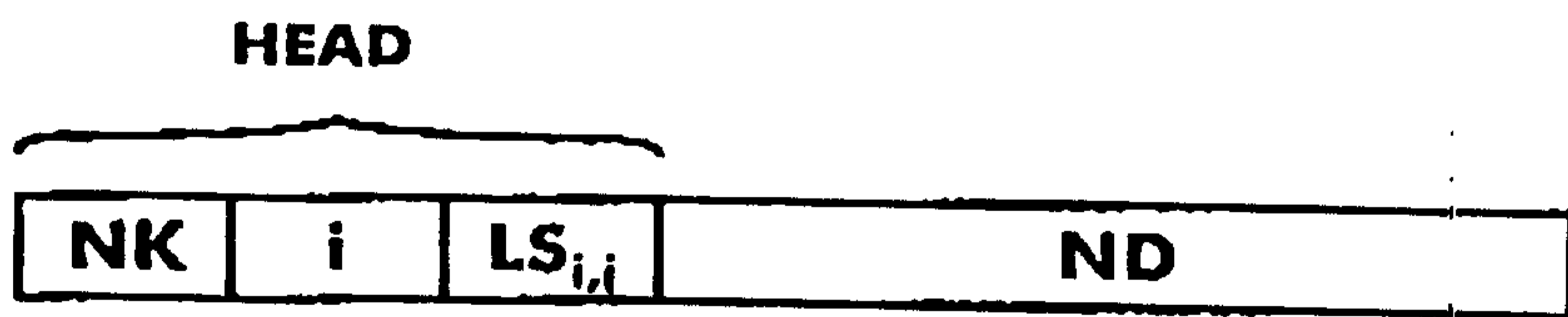


Fig.6

5/6

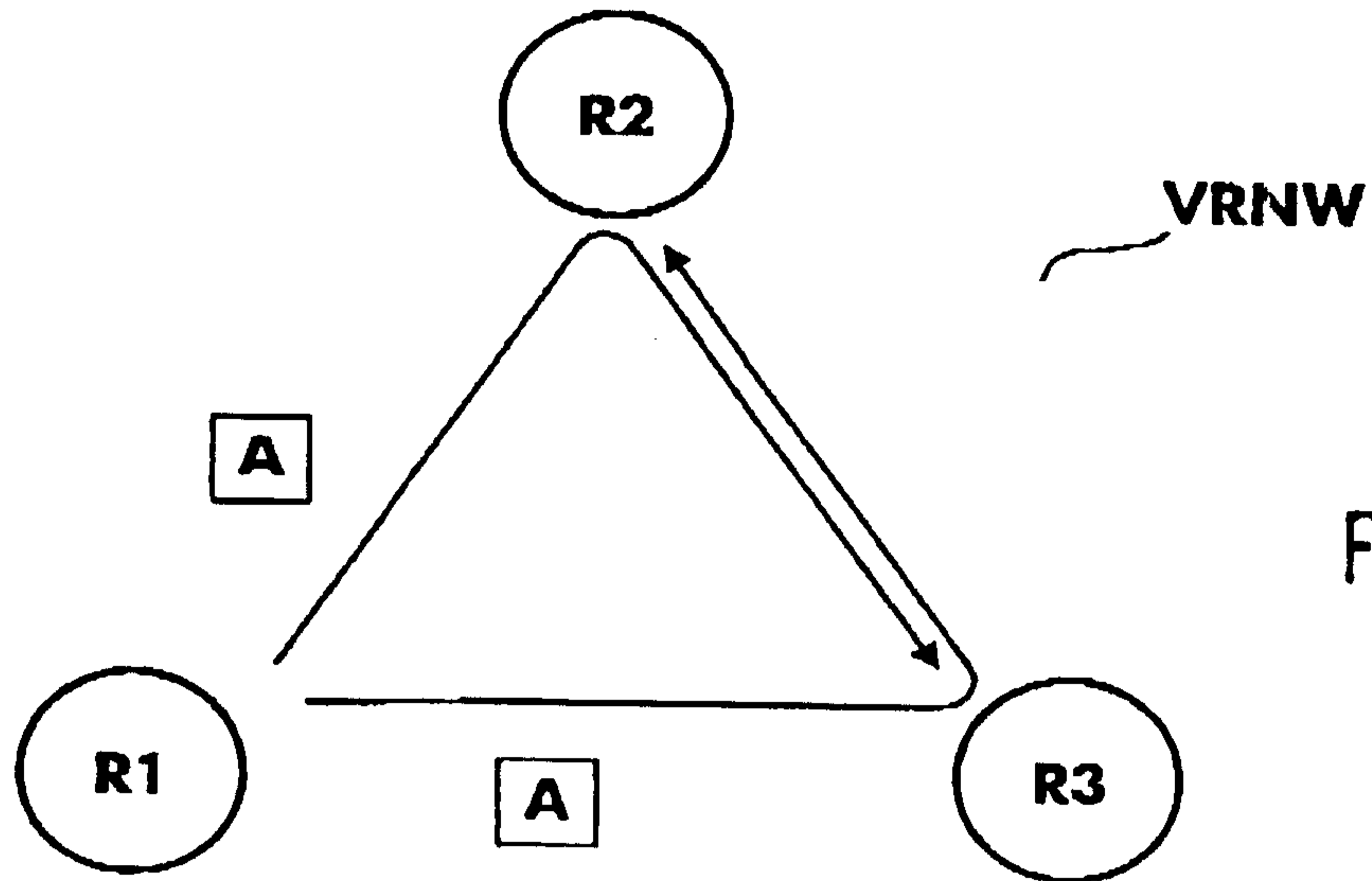


Fig.7

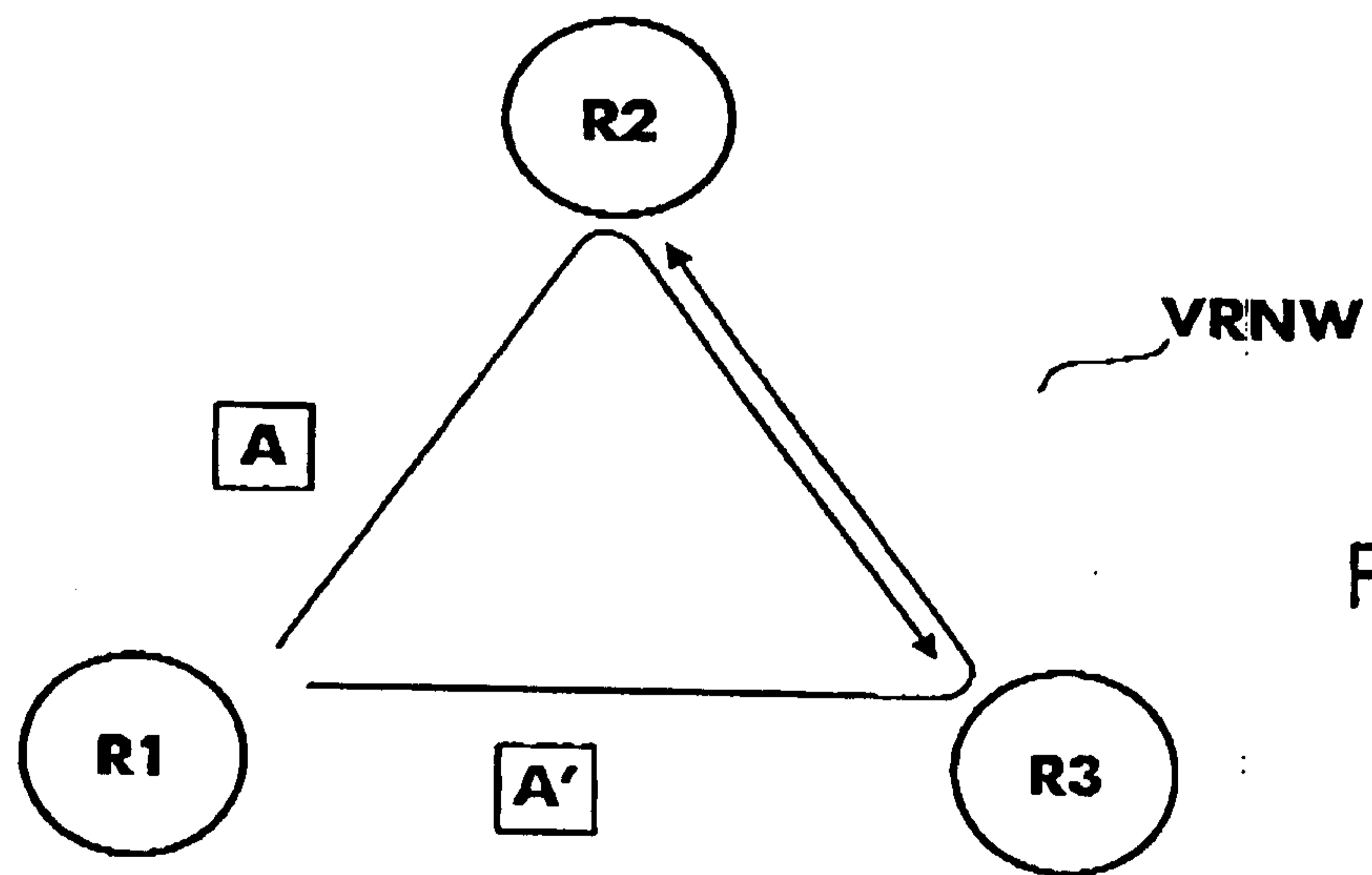


Fig.8

$$\boxed{A} \neq \boxed{A'}$$

6/6

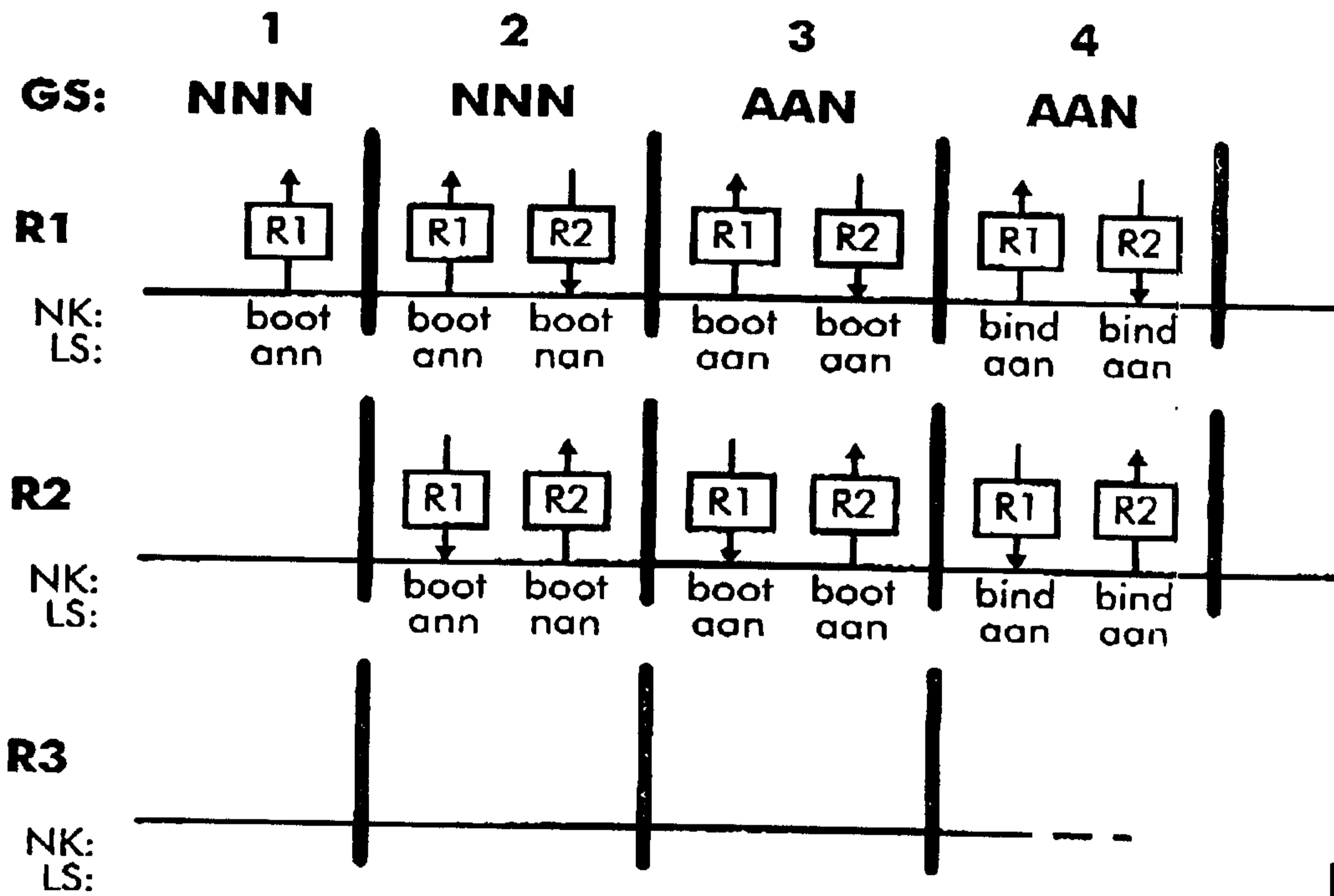


Fig. 9

