TECHNION - Israel Institute of Technology
Computer Science Department

A FORMAL APPROACH TO COMMUNICATION-NETWORK PROTOCOL; BROADCAST, AS A CASE STUDY

by

Baruch Awerbuch* and Shimon Even**

Technical Report #284

June 1983

ABSTRACT

In the area of communication-networks, many protocols have been suggested, and some are in practical use. In the case of networks whose topology continuously changes, no protocol has been proved, since no formal ground rules have been suggested.

We present a mathematical model of such a network, by means of 7 axioms.

A protocol, BBP, for performing broadcast is presented and proved to be reliable, if the network behavior allows reliable broadcast at all. We know of no previous protocol which achieves this goal.

---

* Department of Electrical Engineering

** Department of Computer Science.

1. INTRODUCTION

1.1 The Model

Consider a store-and-forward computer communication network described by an undirected graph G(N,L) where the nodes, N, represent computing units responsible for communication and the links, L, represent bidirectional noninterfering communication channels operating between them. Nodes connected by a link are said to be neighbours. Each node has unbounded processing and memory capabilities and is pre-programmed to perform its part of the computation as well as to receive and send messages to neighbours. These actions are assumed to be performed in zero time. In a fixed topology network, each link in each direction has some finite positive delay which may change with time arbitrarily, subject to the FIFO rule. In other words, each message sent by node i to node j arrives correctly within a finite undetermined time and all messages are received at j in the same order as they have been sent by i.

In the present paper we deal with networks of changing topology, where links may fail and recover again arbitrarily, but nodes never fail. The communication properties of the links of such networks are more complicated than those of fixed topology, and are described in Section 4.

1.2 The Problem

Broadcast is the delivery of copies of a message to all nodes in the communication network. Broadcast messages will be referred to as packets. The most important properties that any "good" broadcast protocol should possess are: reliability, low broadcast cost and low delay. Completeness means that all the packets which are accepted at a node are accepted in the same order as they have been released by the source, without duplication or omission. Finiteness means that every packet is accepted at every node in finite time. Reliability is the combination of completeness and finiteness. The Broadcast Cost of a packet B, BC_B, is the number of times B traverses links of the network.

1.3 Existing Solutions

A survey of existing broadcast algorithms is given by Dalal and Metcalfe in [4,5]. They argue that the most practical broadcast protocol is the "Extended Reverse Path Forwarding" (ERPF). In this protocol, broadcast from a source s is performed along the Routing Structure of s (see Section 2.1) in the reverse direction, and thus no special trees need be maintained for broadcast. However, they show that broadcast in ERPF may not be reliable even if the network's topology is fixed. For the special case of constant topology network and the routing protocol of Merlin & Segall [2], an improved reliable version of ERPF was proposed in [6]. However, this method does not apply to other situations, i.e. changing topology or other routing protocols. In fact, none of the existing protocols achieve reliable broadcast in networks with changing topology. Moreover, no rigorous description of communication properties of such networks is known to us.

1.4 The Contents of this Paper

In Section 2 we present the Basic Broadcast Protocol (BBP). The input to BBP is an arbitrary set of links on which broadcast must be performed. The BBP operates in a network with arbitrarily changing topology where links may fail and recover infinitely many times. The communication properties of such links are axiomatized in Section 4. (Similar assumptions are informally described by Segall [1].) No universal time measurement is assumed, i.e. each node has its own (independent) clock. The properties of BBP are rigorously proved in Section 5 using the axioms presented in Section 4. It is shown that BBP is always complete. Bounds on broadcast cost of BBP are given. The notion of "eventual connectivity" is defined for dynamic structures in Section 3. It is shown that if the input structure of the BBP is eventually connected, then the BBP is finite. Otherwise, no broadcast protocol can be finite.

Since a successful Routing Protocol construct an eventually connected Routing Structure, its choice as an input to the BBP yields a reliable broadcast. Thus, ERPF can be made reliable even in networks with changing topology. It can be shown that other choices of the input yield improved versions of other known broadcast algorithms.

2. THE BASIC BROADCAST PROTOCOL (BBP)

2.1 The Routing Protocol and Routing Structure

A familiarity with the notions of Routing Protocol (RP) and Routing Structure (RS) is useful in order to understand the nature of the input to BBP. Let us describe briefly these notions. For details, see [2,3].

The purpose of RP is to deliver the single-source single-destination messages along "short" paths (in a sense of delay, global network cost, etc.). RP specifies, for every node $i \in N$, time t for this node and every possible destination $s \neq i$, the 'preferred neighbours' set $P_i^s[t]$. A message destined for s, arriving at node i at time t is forwarded by i to a node $j \in P_i^s[t]$. This process is repeated (at j) until the message eventually arrives at s. The set of directed links $P^s[t] = \{(i,j)\ i \in N$ and $j \in P_i^s[t]\}$ for fixed s, t is called the s-th Routing Structure (RS) at time t (see Figure 1).

2.2 The Input to BBP

The input structure F^s is an arbitrary time-varying subset of the directed network's links. It is distributively updated by some external protocol which specifies for each node $i \in N$, time t (at i) and each possible broadcast source $s \neq i$, the set of fathers $F_i^s[t]$, w.r.t. s. It contains the neighbours of i which are in charge of delivery of packets from the broadcast source s to i.

For example, this external protocol might be the RP, i.e. one may choose $F_i^s[t] = P_i^s[t]$ for each i, t, s. Then, BBP will perform reliable broadcast on the links of the s-th RS and can be viewed as a reliable version of ERPF.

Note that the input does not inform node i to which neighbours it should deliver the packets of s, i.e. for which k $i \in F_k^s$ holds. The set $Z_i^s[t]$ of nodes to which node i forwards packets (of s at time t) is called the set of sons of i (w.r.t.'s). It includes those nodes k for which $i \in F_k^s$ holds according to the (possibly, outdated) information i has. This information is obtained using special updating messages.

It is only reasonable to assume that the set of times (at node i) when F_i is changed contains no infinite convergent subsequence.

2.3 Preliminaries

We assume that the packets are all different, so that duplicates can be detected. The basic idea of BBP is that every new packet arriving at a node is accepted, contrary to the operation of ERPF and its improved versions [6,7], where only packets arriving from the father are accepted. Reliability is achieved by introducing additional memory to the nodes. Before describing the protocol, let us explain our notations. The protocol is performed by all nodes i of the network, and each one performs the same node algorithm. The notation "$x_i^s[t]$" means: "variable x kept at node i at moment t with respect to source s". From here on, we assume that there exists a single broadcast source s, (i.e. broadcast processes from different sources do not interfere) and the superscript s will be omitted. We also omit "[t]" when the time in question is clear from the context. When we write: "node j sends message $M(x_j)$ to j at time T" it means that this message contains the value of $x_j[T]$. When this message arrives at i, it is stamped with the identification j of the node it came from and has the format $M(j,x)$, where x is the value of $x_j[T]$.

2.4 Description of the BBP

Each node i is required to keep the following variables;

1) $LIST_i$, where every accepted packet is stored in the received order, since the beginning of the algorithm;

2) IC_i, which keeps count of the number of packets in $LIST_i$;

3) $IC_i(j)$ which is the estimate of IC_j at node i, kept for every neighbour j;

4) F_i set of fathers of node i;

5) Z_i set of sons of node i;

6) E_i set of neighbors of node i.

Now, we present formally the node algorithm. It specifies the actions taken at node i for all possible events, which are either receipt of a message $M(j,x)$ or change in F_i.

"For $M(j,x)$" means: "whenever $M(j,x)$ arrives at i, i performs the following". Failure and recovery of a link (i,j) are represented by receipt of FAIL(j) and WAKE(j) messages at node i.

BBP-Algorithm for node i

F. For change in F_i:

1) <u>if</u> j becomes a new member of F_i <u>then</u> send $DCL(IC_i)$ to j
/* $j \in E_i$ */

2) <u>if</u> k ceases to be a member of F_i, and $k \in E_i$ <u>then</u> send CNCL to k.

D. For DCL(j,IC):

1) $Z_i \leftarrow Z_i \cup \{j\}$; /* recognize j as a son */

2) if $IC_i(j) < IC$ then $IC_i(j) \leftarrow IC$; /* else leave $IC_i(j)$ unchanged */

3) while $IC_i(j) < IC_i$, do:

4) send to j the contents of $LIST_i(IC_i(j) + 1)$;

5) $IC_i(j) \leftarrow IC_i(j) + 1$ od

B. For B(j) /* a packet B arriving from j */

1) if $B \notin LIST_i$ then /* B is new */

2) $IC_i \leftarrow IC_i + 1$;

3) put B in $LIST_i(IC_i)$; /* acceptance */

4) if $j \in Z_i$, $IC_i(j) = IC_i - 1$ then $IC_i(j) \leftarrow IC_i$;

5) for every $k \in Z_i$, $IC_i(k) = IC_i - 1$ do

6) send B to k, $IC_i(k) \leftarrow IC_i$ od.

C. For CNCL(j)

1) $Z_i \leftarrow Z_i - \{j\}$ /* drop j from the list of sons */

W. For WAKE(j)

1) $E_i \leftarrow E_i \cup \{j\}$;

2) $IC_i(j) \leftarrow 0$ /* reset the variables */

FL. For FAIL(j)

1) $E_i \leftarrow E_i - \{j\}$;

2) $F_i \leftarrow F_i - \{j\}$;

3) $Z_i \leftarrow Z_i - \{j\}$.

3. PROPERTIES OF BBP

3.1 Major Properties

We state now the properties of BBP, which hold for the most general conditions, i.e. for arbitrary input structure F and for a network which suffers from an arbitrary (maybe, infinite) sequence of topological changes. These properties are deduced from the axioms presented in Section 4. We first define the concept of "eventual connectivity".

Definitions Denote $\bar{F}_i = \bigcap_t \bigcup_{t' \geq t} F_i[t']$ and $\bar{F} = \{(i,j) | j \in \bar{F}_i\}$, and consider the digraph $G(N,\bar{F})$. This graph contains the links (i,j) which are persistent in F, i.e., reappear after each deletion. F is said to be eventually connected w.r.t. node s if there exists a directed path from every node i to node s in the digraph $G(N,\bar{F})$. The protocol is said "to perform broadcast on links of F" if for every link $(i,j) \notin \bar{F}$, the protocol, at some time, ceases to propagate packets from j to i.

Claim 1 Broadcast is always complete.

Claim 2 Broadcast is finite (and thus, reliable) iff the input structure F is eventually connected w.r.t. the broadcast source s.

Note: According to the definition above, it can be easily shown that no protocol can perform reliable broadcast from s on the links of F if F is not eventually connected w.r.t. s.

Definitions: Let BC_B, V_B, E_B be the number of times packet B traverses the network's links, the number of nodes which accept B, an upper bound on the number of undirected links between nodes which accept B, respectively. Also, let 2D denote an upper bound on the roundtrip delay of a link and let $\frac{1}{\lambda}$ be a lower bound on the time between two non-trivial changes in F_i (of the kind $F_j \leftarrow F_j \cup \{j\}$, $j \neq i$).

Claim 3: Broadcast cost in BBP satisfies:

(a) $BC_B \leq 2E_B - (V_B-1)$

(b) If $|F_i[t]| \leq 1$ for all $i \in N$ and all t, then
$BC_B \leq \min\{2E_B - (V_B-1), (V_B-1)(2 + 2D\lambda)\}$.

(c) If F converges to a (constant) spanning tree, then $BC_B = |N|-1$ holds for all B released after such a convergence ($|N|$ = total number of nodes).

3.2 Additional Interpretation

Observe that a structure F is eventually connected w.r.t. node s iff for every start node and time it is possible to reach node s eventually by means of the following Ideal Routing process:

1) At time t, move from i to $j \in F_i[t]$ in zero time.
2) Upon arrival at intermediate node, wait there for undefinite time and then perform 1).

The Routing Protocol (see Section 2.1) delivers the messages to destination by means of the following Actual Routing process:

(1) At time t, nove from i to $j \in F_i[t]$ in time equal to the delay of the link (i,j) (at time t).
(2) Upon arrival at intermediate node i, wait until the set P_i^s is non-empty and then perform 1).

Clearly, the Actual Routing can be simulated by the Ideal Routing by waiting at the intermediate node for the time equal to the delay of the incoming link.

Let us define a Routing Protocol to be reliable if the Actual Routing process delivers each message in finite time to its destination s. By the above argument, the s-th Routing Structure used by this Routing Protocol must be eventually connected w.r.t. s. Thus, using the s-th Routing Structure as the input to the BBP, i.e. choosing $F_i[t] \leftarrow P_i^s[t]$ for all i, t we achieve the reliable broadcast.

Corollary. If the input to the BBP is the s-th Routing Structure, then if the Routing to s is reliable then the Broadcast from s is reliable. Thus, BBP can be viewed as a reduction from the problem of Broadcast to the problem of Routing.

4. PROPERTIES OF A LINK

Here, we define precisely the communication properties of a link (i,j) in presence of topological changes. These properties we secured by the underlying link - protocol. Less formal assumptions were presented by Segall [1]. We postulate 5 properties A1, A2, A3, A4, A5, A6, A7 and start with their informal description.

A1 says that messages can be sent and received over the link only in some "operating intervals". A2 says that when a link recovers, no messages can be in transit through it. A3 says that the messages sent in the same operating interval obey the FIFO (first in - first out) rule. A4 says that failures of a link are detected in finite time. A5 says that if link (i,j) is operating, there is a "fair chance" that a message sent by i to j will indeed arrive at j, i.e. there is a correspondence between the status of link (i,j) as seen at i and the actual capability of the link to deliver messages to j. Thus, if the link does not fail terminally and i "insists" on delivery of a message to j, it will eventually succeed. A6 says that message travelling in the network cannot return to its start-point before it was sent. A7 says that an unbounded sequence of departure times cannot yield a bounded infinite sequence of arrival times.

The axioms refer to facts as viewed by node i:

A1) <u>Operating intervals</u>: At both ends (i,j) of the link the link - protocol generates alternating sequences of WAKE and FAIL messages, which inform the recoveries and failures of the link. From the point of view of node i the link (i,j) is said to be <u>operating</u> in the closed time interval between receiving WAKE(j) and FAIL(j) messages.

The above interval is called the operating interval (e.g. (t_1, t_1') of Figure 2). Node i can send (receive) messages to (from) node j only when link (i,j) is operating. A message sent by i to j does not necessarily arrive at j.

A2) Communicating intervals: Two operating intervals π, ϕ at opposite sides i, j of the link are said to be communicating if it is possible for node i to send a message to node j at interval π, so that it will be received at j at interval ϕ or vice versa. We denote this relation by $\pi \sim \phi$. If interval π_1 preceeds interval π_2, we write $\pi_1 < \pi_2$. It is postulated that $[(\pi_1 \sim \phi_1) \wedge (\pi_2 \sim \phi_2) \wedge (\pi_1 < \pi_2)] \supset (\phi_1 < \phi_2)$ i.e. communication relation "$\sim$" is monotonous in time.

Example: In Figure 2, $\pi_1 \sim \phi_1$, because messages M_1, M_2 sent by i at times $t_A, t_B \in \pi_1$, arrive at j at times (measured by j) $T_A, T_B \in \phi_1$. We conclude that $\pi_1 \not\sim \phi_0$, $\phi_2 \not\sim \pi_0$, $\phi_1 \not\sim \pi_0$, $\pi_2 \not\sim \phi_0$, etc.

A3) FIFO: Suppose messages A, B are sent by node i to j during the same operating interval. If A is sent before B and B arrives at j, then A arrives too and its arrival time preceeds that of B.

A4) Failures detection: Suppose that in response to a message M received by j from i, j will send to i an "acknowledgement" R. The existence of a constant $2D > 0$ (called "bound on roundtrip delay of a link") is assumed such that in at most $2D$ time-units after M is sent from i to j, i will receive either FAIL or R.

A5) Consider any unbounded sequence of times $s = \{t_k\}_{k=1}^{\infty}$, generated by an on-line algorithm operating at node i such that link (i,j) is operating at each t_k. For any such s, we assume existence of time $t_m \in s$ such that a message sent from i to j at t_m successfully arrives at j.

Comments: We require that s is generated by an on-line algorithm for the following reasons. Otherwise, we might deliberately choose s to be a set of times when link (i,j) is still operating, but is about to fail and no message sent at $t \in s$ from i will succeed in reaching j. If link (i,j) fails many times, then s might be an unbounded set and it seems that A5 is violated. However, this is not the case, because generation of such set s requires information about future failures of the link and thus s cannot be generated by an on-line algorithm. If there were a way to predict the link's failure in advance, the link's protocol should have used this information to declare a FAIL message, to prevent the hopeless transmission of messages which cannot reach their destination.

Before proceeding further with additional axioms, a brief discussion is helpful. In a centralized algorithm, the statement that action A is performed before action B (or in short, A preceeds B) means that the execution of A may influence the execution of B but the outcome of B has no influence on A. Observe that for any A, B either A preceeds B or vice versa. In a distributed algorithm, it may happen that actions A, B are performed concurrently and therefore neither can influence the other. This happens when actions A, B are performed at different nodes and neither can communicate the outcome of its action to influence the action of the other.

In the situation above, where no casuality connection exists between events A, B it is improper to say that "A is performed before B" or vice versa. Also observe that different users of the network might have different time scales and rates (say in an interplanet or interstellar communication) so that no global time clock exists in in the network, and a quantitative comparison of times at different nodes is impossible.

For these reasons, we wish to redefine the "before" relation, and will stick to this definition throughout this paper, unless otherwise stated.

For actions performed in the same node, "before" relation is defined in the usual sense. Now, we define a new "before" relation and show that is is an extension of the usual "before" relation, in the sense that the new relation contains the old one.

Definition: Action A is said to be performed "before" action B if the outcome of A can reach the node which performs B before the execution of B.

We denote this by $t_A < t_B$, where t_A, t_B denote the times when actions A, B are performed, as measured in the respective nodes.

Observe that the "before" relation defined above is transitive. Also it constitutes an extension of the old "before" relation in the usual sense, defined for events happenning at the same node, because a node "delivers" messages to itself in zero time.

A6) The relation "before" is irreflexive.

Discussion: If $t_1 < t_2$ then A6 implies that $t_1 \neq t_2$. Also, by transitivity of "before" this implies that $t_2 \nless t_1$.

The purpose of A6 is to preserve the usual sense of "before", when times are compared in the same node. Let t_1, t_2 be times of events in the same node. If $t_1 < t_2$, in the new sense, then $t_1 < t_2$ in the old sense too, but $t_2 \nless t_1$ by the discussion above. Thus, the new "before", when restricted to times of events in one node yield the old "before" for that node.

<u>A7)</u> For any 2 infinite sequences of times $s = \{t_k\}_{k=1}^{\infty}$ and $S = \{T_k\}_{k=1}^{\infty}$, if $T_k < t_k$ for all k and S is unbounded then s is unbounded too.

5. FORMAL ANALYSIS OF BBP

Here, we prove formally the properties of BBP. In Sections B.1, B.2, B.3, we prove Claims 1,2,3 (Completeness, Finiteness, Broadcast Cost) stated in Section 3. In B.1, we use only assumption A3, A6, A7. In B.2, B.3 we use all the assumptions.

B.1 Completeness

To simplify the proofs, we shall modify the original version of BBP, which will be referred to as BBP1, and the modified version will be referred to as BBP2. We prove that BBP2 is complete and equivalent to BBP1.

Let the packets be numbered in the order which the source releases them. We denote the counter-number of packet B by IC(B). In BBP2 it is assumed that every packet B contains IC(B). The procedure which handles an arriving packet is now modified in BBP2 as follows:

B. For B(j) /* a packet B arriving from j */

1) <u>If</u> $IC(B) > IC_i$, <u>then</u> /* B is new */

2) $IC_i \leftarrow IC(B)$

3) put B in $LIST_i(IC_i)$ /* acceptance */

4) <u>if</u> $j \in Z_i$, $IC_i(j) < IC_i$, <u>then</u> $IC_i(j) \leftarrow IC_i$

5) <u>for</u> every $k \in Z_i$, $IC_i(k) < IC_i$ <u>do</u>

6) send B to k, $IC_i(k) \leftarrow IC_i$.

Now, we prove that BBP2 is complete, We need first some preliminary results.

For the sake of brevity let us introduce the following convention: By "condition P holds at time t^+" we mean that there exists an $\epsilon > 0$ such that for every time t' in the open interval $(t, t+\epsilon)$ the condition P holds. Similarly, t^- is defined.

In the following proofs we shall denote the line labeled x of algorithm BBP2 by <x>.

Lemma B.1.1

(a) $IC_i[t]$ is nondecreasing with t.

(b) $IC_i(j)[t]$ is nondecreasing with t while link (i,j) operates.

(c) If a packet B is accepted by node i at t, (i.e. <B3> is performed) then $IC[t^+] = IC(B)$.

(d) If a packet B arrives at node i at t then for all $t' > t$ $IC_i[t'] \geq IC(B)$.

(e) The only location in which B is ever stored in $LIST_i$ is IC(B).

(f) If a packet B is sent from i to j at t, then $IC_i(j)[t^+] = IC(B)$.

(g) If at time t, node i does not perform any action of BBP2 and $j \in Z_i[t]$ then $IC_i(j)[t] \geq IC_i[t]$.

Proof of Lemma B.1.1

The proof of the above claims, one by one, is straightforward. For each claim we indicate the lines of BBP2 and the previous claims which imply it.

(a) <B1>, <B2>.

(b) <W2>, <D2>, <D5>, <B4>, <B5>, <B6>.

(c) <B1>, <B2>.

(d) <B1>, <B2>, (a).

(e) <B2>, <B3>

(f) <D4>, <D5>, <B2>, <B6>,

(g) <D3>, <D5>, <B4>, <B5>, <B6>, <FL3>, <W2>.

(FAIL preceds WAKE and therefore on WAKE(j), $j \notin Z_i$.)

Q.E.D.

Lemma B.1.2

As long as completeness of BBP2 is maintained at node i, it actually performs the actions of BBP1, i.e.

(a) In <B1>, $IC(B) > IC_i$ implies $IC(B) = IC_i + 1$ and thus in <B2>, $IC_i \leftarrow IC_i + 1$ is performed.

(b) In <B4> and <B5>, $IC_i(k) < IC_i$ implies $IC_i(k) = IC_i - 1$ for all sons $k \in Z_i$, and thus in <B4> and <B6> $IC_i(k) \leftarrow IC_i(k) + 1$ is performed.

Proof:

(a) follows from the definition of completeness.

(b) follows from (a) and Lemma B.1.1 (g).

Q.E.D.

Theorem B.1.1

Suppose a message M is sent from j at time T and arrives at i at time t. Then:

(a) $IC_j(i)[T^-] \leqslant IC_i[t^-]$

(b) $IC_j(i)[T^+] \leqslant IC_i[t^+]$.

Proof of Theorem B.1.1

If M is not a broadcast packet, then the variables mentioned in the theorem do not change at times T, t (respectively) and thus (a) is equivalent to (b). Otherwise (M is a broadcast packet), then (b) holds by Lemma B.1.1 [(d),(f)], because $IC_i[t^+] \geqslant IC(M) = IC_j(i)[T^+]$.

Thus, it is sufficient to prove (a). Let us denote $x = IC_j(i)[T^-]$ and prove that $x \leq IC_i[t^-]$. Denote by T_1 the last time before T when $IC_j(i) \leftarrow x$ was performed. If $x = 0$, the claim is trivial. Assume $x > 0$. At time T_1 the link (i,j) operates, and at time T it operates too. It could not have failed and waked in between, since in <W2> $IC_j(i) \leftarrow 0$ is performed.

At T_1, one of the following events could happen:

(1) j receives DCL(i,x) with $x > IC_j(i)[T_1]$ (<D2>)

(2) B is accepted at j from i and $i \in Z_j[T_1]$, $IC_j(i)[T_1] < x$ (<B4>)

(3) B is sent from j to i (<D4>, <B6>).

In cases (1), (2) denote by t_o the time when node i sent the above DCL or B, respectively. Clearly, $t > T > T_1 > t_o$ and by transitivity, $t > t_o$. Thus, $t^- > t_o$ and by B.1.1(a)
$IC_i[t^-] \geq IC_i[t_o] \geq IC(B) = x$.
In case (3), by A3, applied to B and M, B arrives at i at time $t_1 < t$. Thus, $t_1 < t^-$. By Lemma B.1.1 [(a), (d)]
$IC_i[t^-] \geq IC_i[t_1^+] \geq IC(B) = x$.

Q.E.D.

<u>Lemma B.1.3</u>

For any event A which happens, at least once, at various nodes of the network, it is possible to find a time t when A happens <u>for the first time</u> in the network in the following sense: For any node n and time t', t' < t, A has not happenned at n until t'.

<u>Proof</u>:

For those nodes n where A happens at least once, denote by t(n) the first time when it happenned at n. Now, pick any such node

n_1. Either $t(n_1)$ satisfies the requirement of the lemma or there exists a node n_2 with $t(n_2) < t(n_1)$.

Applying repeatedly the avove construction, we will eventually stop after a finite number of steps, finding the required node and time. The fact that the above process indeed stops and cannot continue infinitely is proved as follows.

Suppose that there exists an infinite sequence of times $t(n_1) > t(n_2) > t(n_3) > \ldots$. The network is finite and therefore there exist integers $m < k$ such that $n_m = n_k$. Thus, both $t(n_m) = t(n_k)$ and $t(n_m) > t(n_k)$; hold. A contradiction to A6.

Q.E.D.

Theorem B.1.2

Broadcast in BBP1 and BBP2 is always complete.

Proof: It suffices to show, by Lemma B.1.2, that in BBP2 completeness is never violated. Assume the contrary, and consider a node i and time t when completeness is violated for the first time in the network in the sense of Lemma B.1.3. Thus, at time t, a "gap" is created in $LIST_i$, i.e. node i receives (and therefore accepts) some packet B with $IC(B) > IC_i[t^-] + 1$.

Suppose B was sent by node j at time T, $T < t$. By our assumption, completeness was maintained at j until T, and the desired contradition follows:

$$IC(B) = IC_j(i)[T^+] = IC_j(i)[T^-] + 1 \leq IC_i[t^-] + 1.$$

The above relations (from left to right) follow from Lemma B.1.1(f), Lemma B.1.2(b), Theorem B.1.1(a).

Q.E.D.

B.2 Finiteness

Definitions:

Denote by V_1^* the set of nodes which accept every packet B in finite time and by V_1^F the set of nodes i such that there exists a directed path from i to s in $G(N,\bar{F})$.

Observe that broadcast is finite iff $V_1^* = N$. By definition of Section 3.1 F is eventually connected w.r.t. s iff $V_1^F = N$. Our purpose is to show that $V_1^* = V_1^F$ for the BBP.

Analogously to the definition above, one can define $\bar{x}$, the set of persistent links of x, for any structure $\{(i,j) | j \in x_i\}$ induced by sets $x_i[t]$ defined for all i, t. Also, define $V_2^* = N - V_1^*$. $V_2^F = N - V_1^F$. Saying that link (i,j) operates after t, we mean that it never fails after t.

Lemma B.2.1

For any structure x, $(i,j) \in \bar{x}$ if and only if one of the following conditions holds:

(a) there exists t_0 such that $j \in x_i[t]$ for all $t \geq t_0$,

(b) there exists an unbounded sequence of times $s = \{t_k\}$, such that j joins x_i at each t_k.

Proof: It is easy to see that if $(i,j) \notin \bar{x}$, then neither (a) nor (b) can hold.

Suppose that $(i,j) \in \bar{x}$. If the set $s = \{t | x_i \leftarrow x_i \cup \{j\} \text{ at } t\}$ is unbounded, then (b) holds. Otherwise, any upper bound t_0 ($t_0 > t$ for all $t \in s$) must satisfy (a).

Q.E.D.

Lemma B.2.2

Suppose that link (k,r) operates after time t_k^o. Then there exist times t_k^* at node k and t_r^* at node r such that:

(a) Links (k,r) and (r,k) operate after times t_k^*, t_r^* respectively, and $t_k^* \leq t_k^o$.

(b) Each message sent by k to r after t_k^* successfully arrives at r after t_r^*.

(c) Each message received by k from r after t_k^* was sent by r after T_r^*.

Proof: Pick any unbounded set $S_k = \{t_k^i\}$ of times at node k such that $t_k^i > t_k^o$, and send at time t_k^i (an imaginary) message M^i from k to r. By A4, each M^i arrives successfully at r at some time t_r^i, and by A2 these times belong to the same operating interval. By A7, the set $S_r = \{t_r^i\}$ is unbounded.

Now, denote by π, ϕ the operating intervals at nodes k, r containing sets S_k, S_r and by t_k^*, t_r^* the start times of these intervals respectively (which exist because, by A1, the intervals are closed).

The intervals π, ϕ contain unbounded sequences of times S_k, S_r and thus are infinite; i.e. links (k,r) and (r,k) operate after times t_k^* and t_r^* respectively. Also, t_k^* is the start time of π and thus $t_k^* \leq t_k^o$, proving (a).

By A4, messages sent by k to r after t_k^* successfully arrive at r. Observe that by construction, π and ϕ are communicating intervals. By A2, messages sent by k at $t \in \pi$ can reach r only at $T \in \phi$ and vice versa, proving (b), (c).

Q.E.D.

Lemma B.2.3

Suppose that a sequence of times $s = \{t_k\}_{k=1}^{\infty}$ satisfies the requirements of A5. Then, it contains an unbounded subsequence $s' = \{t_{k_m}\}_{m=1}^{\infty}$ such that for each m a message sent from i at t_{k_m} successfully arrives at j.

Proof: By A5, such subsequence s' contains at least one member t^1. The truncation s_1 of s, defined as $s_1 = \{t | t \in s, t > t^1+1\}$ still satisfies the condition of A5, and thus s' contains some member $t^2 \in s_1$; $t_2 > t_1+1$. Repeatedly continuing with the above argument, we see that s' is unbounded.

Q.E.D.

Theorem B.2.1

$(j,i) \in \bar{F}$ iff $(i,j) \in \bar{Z}$.

Proof of Theorem B.2.1 By Lemma B.2.1, it is sufficient to prove the two following claims.

Claim 1: There exists time t_o at i such that $j \in F_i[t]$ holds for all $t \geqslant t_o$ if and only if there exists time T_o at j such that $i \in Z_j[T]$ for all $T \geqslant T_o$.

Claim 2: There exists an unbounded sequence $s = \{t_k\}_{k=1}^{\infty}$ of times when i performs $F_i \leftarrow F_i \cup \{j\}$ if and only if there exists an unbounded sequence $S = \{T_k\}_{k=1}^{\infty}$ when j performs $Z_j \leftarrow Z_j \cup \{i\}$.

Proof of Claim 1: The "only if" part

Suppose that such time t_o exists. The set of times when F_i changes contains no cluster points, as assumed in Section 3.1. Thus, there exists the time t_1 when $F_i \leftarrow F_i \cup \{j\}$ is performed for the last time. For all $t \geqslant t_1$, link (i,j) operates; because if link

(i,j) ever fails after t_1 then $F_i \leftarrow F_i - \{j\}$ (see <FL2>) would have been performed, in contradiction to the assumption that $j \in F_i[t]$ for $t \geq t_1$.

Observe that at t_1 the last DCL(i,·) has been sent by i to j, and no CNCL(i) is ever sent by i to j after t_1.

By Lemma B.2.2 [(a),(b)] this DCL message successfully arrives at j at some time T_1, after which the link (i,j) operates (i.e. $T_1 \geq t_j^*$ in terms of Lemma B.2.2). Moreover, no CNCL(i) can arrive to j after T_1 by A2, A3. Thus, $i \in Z_j[T]$ for all $T \geq T_1$, implying $i \in \bar{Z}_j$.

Q.E.D.

The "if" part

Suppose that such time T_0 exists. The set S of departure times of DCL(i,·) messages sent from i to j contains no cluster points, as assumed in Section 3.1. By A7, the set of arrival times does not contain any cluster points either.

Thus, there exists a time T_1 when $Z_j \leftarrow Z_j \cup \{i\}$ is performed for the last time.

For all $T \geq T_1$, link (j,i) operates, because if link (j,i) ever fails after T_1 then $Z_j \leftarrow Z_j - \{i\}$ (see <FL3>) would have been performed in contradiction to the assumption that $i \in Z_j[T]$ for $T \geq T_1$.

Observe that at T_1, the last DCL(i,) is received by j and no CNCL(i) arrives at j after T_1.

By Lemma B.2.2 [(a),(c)] the above DCL is sent by i at such time t_1, after which link (i,j) operates i.e. $t_1 \geq t_i^*$ in terms of Lemma B.2.2. Moreover, no CNCL(i) can be sent from i to j after t_1

by A2, A3. Thus, $j \in F_i[t]$ for all $t \geq t_1$, implying $j \in \bar{F}_i$.

Q.E.D.

Proof of Claim 2 The "only if" part

At each t_k, a DCL(i,•) message is sent from i to j. The set $s=\{t_k\}$ satisfies the condition of Lemma B.2.3 and thus, there exists an infinite subsequence of DCL(i,) message which succeed in reaching j, whose departure times constitute an unbounded subsequence of s. By A7, the sequence S of their arrival times is unbounded and each arrival causes j to perform $Z_j \leftarrow Z_j \cup \{i\}$. Thus, $i \in \bar{Z}_j$.

Q.E.D.

The "if" part:

At each T_k, a DCL(i,•) message is received by j from i. Denote by t_K the time when this message was sent from i. Clearly, $F_i \leftarrow F_i \cup \{j\}$ was set at t_k, and $s = \{t_k\}$ is infinite set of times which, by assumption of Section 3.1, contains no cluster points. But then s is necessarily unbounded and thus $j \in \bar{F}_i$. Q.E.D.

Q.E.D. for Theorem B.2.1

Theorem B.2.2

If $(j,i) \in \bar{Z}$ then every packet B accepted at j at time T_B is also accepted by i at some finite time t_B. In particular, $j \in V_1^*$, implies $i \in V_1^*$.

Proof of Theorem B.2.2

$(j,i) \in \bar{Z}$ if $i \in \bar{Z}_j$ and thus there exists a sequence of times $S = \{T_k\}_{k=1}^{\infty}$ satisfying the premise of A5 such that $i \in Z_j[T_k]$. Clearly, T_k can be chosen so that at every moment T_k, j does not perform any action of BBP (this follows from the fact that the line (i,j) operates at T_k^+) and that $T_k > T_B$, (because the truncation S_1 of the set $\{T_k\}$ to $T_k > T_B$, still satisfies the premise of A5). Suppose that for every k we send (an imaginary) message M_k at time T_k. By A5, there exists a message M_q which arrives at i at some time t_q. Then, B is accepted at i before t_q because: $IC_i[t_q] \geq IC_j(i)[T_q] \geq IC_j[T_q] \geq IC_j[T_B^+] \geq IC(B)$.

The above inequalities are implied by (from left to right): Theorem B.1.1, Lemma B.1.1 (g), Lemma B.1.1 (b), Lemma B.1.1 (c).

Q.E.D.

Theorem B.2.3

In BBP, $V_1^F = V_1^*$.

Proof: Is is sufficient to prove the 2 following claims:

Claim 1: $V_1^F \subset V_1^*$.

Calim 2: $V_1^* \subset V_1^F$.

Proof of Claim 1: Proceeds by induction on the length d_j of the shortest directed path in $G(V,\bar{F})$ from a node $j \in V_1^F$ to the node s.

Here the induction step is the Theorem B.2.2 and the induction basis is the obvious claim: "$s \in V_1^*$". Q.E.D.

Proof of Claim 2: By definition of V_1^F and V_2^F, for any $i \in V_2^F$, $j \in V_1^F$, holds $j \notin \bar{F}_i$ (otherwise, $i \in V_1^F$). By Theorem B.2.1, $i \notin \bar{Z}_j$, and thus there exists time $T(j,i)$ such that after it $i \notin Z_j$ holds and thus j will not forward any packet to i.

Now, consider the packet B such that

$$IC(B) = \max\{IC_j[T(j,i)] \mid i \in V_2^F \;\; j \in V_1^F\} .$$

Clearly, no packet B' with $IC(B') > IC(B)$ can be forwarded to nodes of V_2^F by the nodes of V_1^F. But $s \in V_1^F$ and thus for any $i \in V_2^F$, node i never accepts such a packet B' and therefore $i \in V_2^*$. This implies that $V_1^* \subset V_1^F$.

Q.E.D.

B.3 Broadcast Cost

Theorem B.3.1

In BBP, for every packet B and nodes $i,j \in V$ the following hold:

(a) Packet B cannot arrive from node j at node i more than once.

(b) If B is accepted at j from i, then B is never sent back to i.

(c) $BC_B \leq 2E_B-(V_B-1)$ with notations of Claim 3 (Section 4) where BC_B, V_B, E_B are as defined in Section 3.

Lemma B.3.1

Suppose that for some time T_1 and nodes j, i holds:

(*) for all $T_3 > T_1$, $i \in Z_j[T_3]$ implies $IC_j(i)[T_3] \geq IC(B)$, then B is never sent from j to i after T_1.

Proof of Lemma B.3.1

Assume B is sent from j to i at time T. Thus $i \in Z_j[T]$. Also, by lines <D5>, <B6>, of BBP, $IC_j(j)$ is incremented by 1 at time T, and by Lemma B.1.1(f) $IC_j(i)[T^+] = IC(B)$. Thus, $IC_j(i)[T^-] = IC(B)-1$. By (*) $T \leq T_1$, and our claim follows.

Q.E.D.

Proof of (a): Consider the first arrival of packet B at node i from node j. Suppose that this copy of B was sent by j at time T_1 and arrived at i at time t_1. It suffices to prove that the premise of Lemma B.3.1 holds for T_1, i, j.

Assume T_3 satisfies $T_3 > T_1$ and $i \in Z_j[T_3]$. If link (j,i) operates during the interval $[T_1,T_3]$ then

$IC_j(i)[T_3] \geqslant IC_j(i)[T_1] = IC(B)$ holds by Lemma B.1.1 [(b),(f)] and we are done. Otherwise, during the interval $[T_1,T_3]$ the link (j,i) fails and then recovers at least once.

Consider the last D = DCL(i,IC) message, which has arrived at j before T_3. Suppose it was sent from i at t_2 and arrived at j at T_2, $T_2 < T_3$. Since at T_3 $i \in Z_j$ and after FAIL(i) $i \notin Z_j$, the link (i,j) must not have failed during the interval $[T_2,T_3]$, i.e. it operates during this interval. Therefore, $T_1 < T_2 < T_3$ must hold (see Figure 3a). Suppose that the times mentioned above belong to the following operating intervals: $T_1 \in \psi_1$, $T_2,T_3 \in \psi_2$, $t_1 \in \pi_1$, $t_2 \in \pi_2$. Then $\pi_1 \sim \psi_1$, $\pi_2 \sim \psi_2$, $\psi_1 < \psi_2$. By A2, $\pi_1 < \pi_2$ and thus $t_1 < t_2$. Then one can deduce that:

$IC_j(i)[T_3] \geqslant IC_j(i)[T_2] \geqslant IC = IC_i[t_2] \geqslant IC_i[t_1^+] \geqslant IC(B)$.

The above relations (from left to right) follow from:

Lemma B.1.1(b), <D2>, <F1>, Lemma B.1.1(a), Lemma B.1.1(d). Thus, one can apply Lemma B.3.1;

Q.E.D. for (a).

<u>Proof of (b)</u>: Suppose that B was sent by i at t_1 and accepted by j at T_1. Clearly, B had not been known at j before T_1 and therefore could not have been sent from j before T_1. It suffices to show that the premise of Lemma B.3.1 holds for T_1, j, i.

Pick any time T_3 with $T_3 > T_1$ and $i \in Z_j[T_3]$ and find times T_2, t_2 and D = DCL(i,IC), the last declared message as in the proof of (a). Thus, during the interval $[T_2,T_3]$ $i \in Z_j$ and link (j,i) operates. Now, two cases are treated separately:

1) $T_2 \leqslant T_1$ (see Figure 3b.1)

Since link (j,i) operates at interval (T_2,T_3), it also operates during interval (T_1,T_3). Then,

$IC_j(i)[T_3] \geq IC_j(i)[T_1^+] \geq IC_j[T_1^+] \geq IC(B)$.

The above relations follow (from left to right) by the Claims (b), (g), (d) of Lemma B.1.1. This completes the proof, in this case.

2) $T_2 > T_1$ (see Figure 3b.2).

Then by A2, A3, $t_1 < t_2$ holds and thus

$IC_j(i)[T_3] \geq IC_j(i)[T_2^+] \geq IC = IC_i[t_2] \geq IC_i[t_1^+] \geq IC(B)$,

by reasons as in the proof of (a).

This completes the proof of the second case and of part (b) of Theorem B.3.1

<u>Proof of (c):</u> Note that V_B is the number of nodes which accept B and E_B is the number of possible undirected links which connect them. The packet B can traverse each such link (i,j) at most once in each direction, (by (a)) and for every node i there exists at least one link (i,k) through which B is never sent back (by (b)). This completes the proof of part (c) of Theorem B.3.1, and of the theorem itself.

Q.E.D.

<u>Theorem B.3.2</u>

If $|F_i[t]| \leq 1$ for all i, t then $BC_B \leq (N-1)(2+2D\lambda)$ (with N, 2D, λ as defined in Section 3).

<u>Proof:</u> It is sufficient to show that for every node i and packet B, i can receive at most $2+2D\lambda$ copies of B. Suppose that exactly r copies, $B^1,\ldots,B^r$, of a packet B have arrived at node i. We have to show that $r \leq 2+2D\lambda$.

For $m = 1,\ldots r$, suppose that B^m was sent to i by a node k_m, $k_m \neq i$, at time T_B^m and arrived at i at time t_B^m. Consider the last message $D^m = DCL(i,IC^m)$ which arrived at k_m from i before T_B^m. Denote by t_D^m, T_D^m the times of its departure from i and its arrival at k_m, respectively. Rearrange, if necessary, the indices $m = 1,\ldots,r$ so that the sequence $t_D^1, t_D^2 \ldots t_D^r$ is increasing. Observe that $t_D^r < t_B^n$ for all n, because

$IC_i[t_D^r] = IC^r \leqslant IC_{k_r}(i)[T_D^{r^+}] \leqslant IC_{k_r}(i)[T_B^{r^-}] = IC(B)-1 < IC(B) \leqslant IC_i[t_B^{n^+}]$.

Since during the interval $[t_D^2, t_D^r]$ at most $1+\lambda(t_D^r - t_D^2)$ DCL messages can be sent from i (by definition of λ),

$r \leqslant 1 + (1+\lambda(t_D^r - t_D^2))$.

By the inequality proven above, $t_D^r < t_B^1$. Thus

$r \leqslant 2 + \lambda(t_B^1 - t_D^2)$,

and it remains to show that $t_B^1 - t_D^2 \leqslant 2D$.

Let $j = k_1$. Observe that by the definition of T_D^1, during the interval (T_D^1, T_B^1) the link (j,i) operates, and by A2 the link (i,j) operates during the interval (t_D^1, t_B^1). If link (i,j) ever fails at interval (t_D^1, t_D^2+2D) then we are done, because $t_B^1 < t_D^2 + 2D$ holds. Otherwise, link (i,j) operates during (t_D^1, t_D^2+2D) and, by our assumption that $|F_i[t]| \leqslant 1$, node i must have sent a CNCL(i) message to j at some time $t_c^1, t_D^1 < t_c^1 \leqslant t_D^2$ (when $F_i \leftarrow F_i - \{j\}$ was performed).

Let us assume that if an when CNCL(i) arrives at j, a confirmation message CC(j) is sent immediately back to i. Since no failure of link (i,j) occurs during (t_c^1, t_c^1+2D), one deduces from A4 that both CNCL(i) and CC(j) arrive successfully at j and i, respectively. Denote the times of their arrivals by T_c^1 and t_{cc}^1. Also,

$t_{cc}^1 < t_c^1 + 2D$.

Clearly, B^1 was sent from j to i during the interval (T_D^1, T_C^1). By A2, link (j,i) operates in this interval. By A3, B^1 arrives at i before $CC(j)$, i.e., before t_{cc}^1. Thus,

$$t_B^1 - t_D^2 < t_{cc}^1 - t_D^2 < t_c^1 + 2D - t_D^2 \leq 2D.$$

Q.E.D.

Corollary:

If $|F_i[t]| \leq 1$ holds for all i, then

$$BC_B \leq \min\{2E_B - (V_B - 1), (N-1)(2D\lambda + 2)\}.$$

The corollary follows from Theorems B.3.1 and B.3.2.

ACKNOWLEDGEMENT

The authors would like to thank Prof. A. Segall for proposing the subject and his valuable help during the earlier stages of this research.

REFERENCES

[1] A. Segall: "Distributed Network Protocols". EE Pub. No.414, July 1981. (To appear in IEEE Trans. on Inf. Theory, January 1983).

[2] P. Merlin and A. Segall: "Failsafe Distributed Routing Protocol", IEEE Trans. on Comm., Vol. COM-27, pp. 1280-1288, September 1979.

[3] E. Gafni and D.P. Bertsekas: "Distributed Algorithms for Generating Loop-Free Routes in Networks with Frequently Changing Topology", IEEE Trans. on Comm., Vol, COM-29, pp. 11-18, January 1981.

[4] Y.K. Dalal and R. Metcalfe: "Reverse Path Forwarding of Broadcast Packets", CASM, Vol. 21, No.12, pp. 1040-1048, December 1978.

[5] Y.K. Dalal: "Broadcast Protocols in Packet-Switched Computer Networks", Ph.D. Thesis, Stanford University, April 1977, DSL Tech. Report 128.

[6] A. Segall and B. Awerbuch: "A Reliable Broadcast Protocol". (To appear in IEEE Trans. on Comm.)

[7] B. Awerbuch: "Reliable Broadcast Algorithm", M.Sc. Thesis, EE Dept., Technion, Haifa, Israel, August 1981 (in Hebrew).

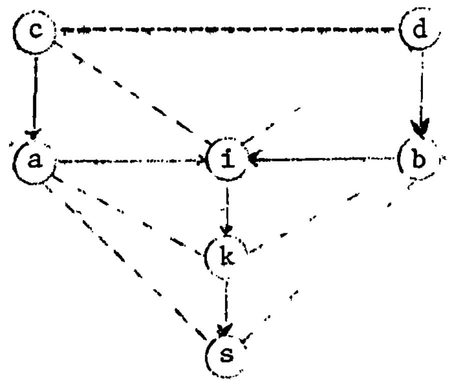


Figure 1 - Routing structure for s.

$P^s = \{(i,P_i^s)\}=\{(c,a),(d,b),(a,i),(b,i),(i,k),(k,s)\}$

(e.g. $P_i^s = k$).

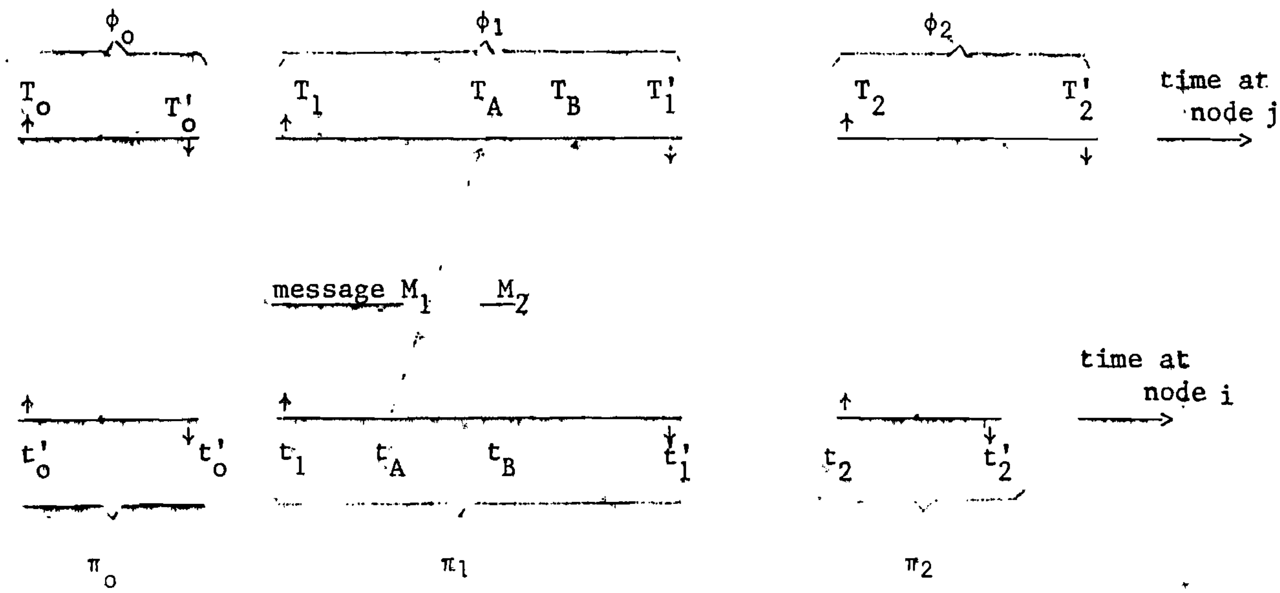
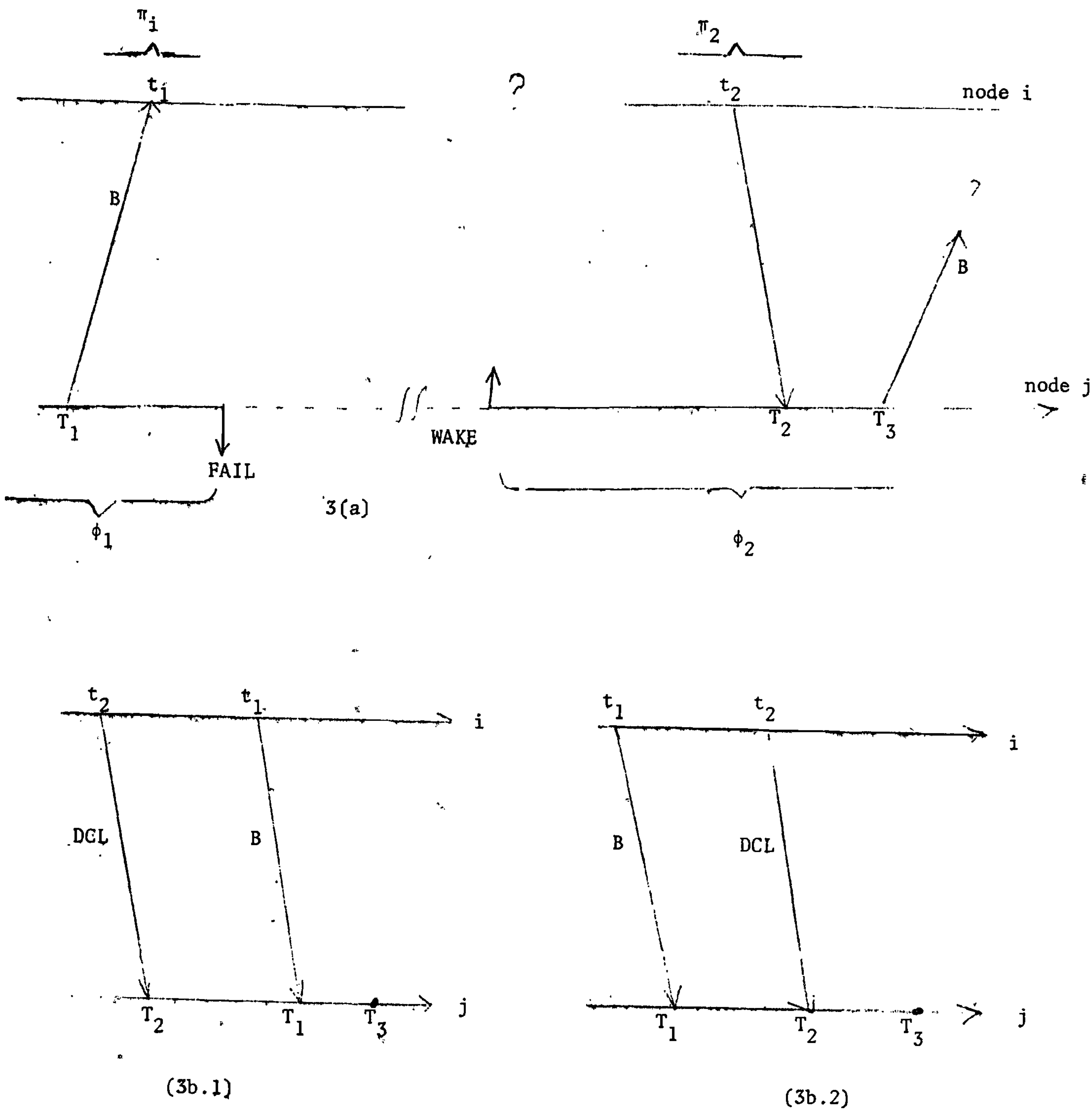


Figure 2 - Link's Operation

↑ - WAKE

↓ - FAIL

↑——↓ - Operating interval



<u>Figure 3</u>: Timing diagram for probing bound on Broadcast Cost.