ソフトウェア基礎論　23−7
(1987. 12. 10)

分散ネットワークにおけるリンク故障診断について

Fault Diagnosis of Links in Distributed Networks

増澤　利光　　　萩原　兼一　　　都倉　信樹
Toshimitsu MASUZAWA　　Ken'ichi HAGIHARA　　Nobuki TOKURA

大阪大学　情報処理教育センター
Education Center for Information Processing
Osaka University

大阪大学　基礎工学部
Faculty of Engineering Science
Osaka University

あらまし　ネットワークの任意の一つのプロセッサが自分に接合するリンクのうち任意の一つが故障しているかどうかを診断する問題（リンク故障診断問題）について考察する．まず，リンクの伝搬遅延に上限がないとき，この問題が解けないことを示す．また，同期式ネットワーク（プロセッサ間に同期があり，リンクの伝搬遅延に上限がある）では，ネットワーク全体での故障プロセッサ数，故障リンク数にかかわらず，この問題が解けることを示す．次に，プロセッサ間に同期はないが，リンクの伝搬遅延に上限がある場合について考察する．この場合，リンク故障診断問題が解けるかどうかは，ネットワークに関するどのような大域情報（隣接プロセッサの識別子，プロセッサ数，辺連結度など）を各プロセッサで利用できるかによって異なることを示し，解けるための条件（プロセッサやリンクの故障状況）について考察する．

Abstract　This paper considers the problem that a processor tests whether the designated link connected to itself is faulty or not. The first result is that no distributed algorithm can solve the problem, when communication asynchrony is assumed. It is also showed that the problem can be solved in synchronous networks regardless of any number of faulty processors and links. Moreover, it is extensively examined how the global information a priori known to each processor such as the topology, the size and the edge-connectivity of the network affects on the solvability of the problem in networks where communication synchrony is assumed but processor synchrony is not assumed.

1．まえがき

ある問題を解くのに必要な情報が，ネットワークで結合された複数台のプロセッサ（以下，PE と略記する）に分散している状況で，それらの情報を交換しながらその問題を解くアルゴリズムを分散アルゴリズムとよぶ．これまでに，最大値探索分散問題や生成木構成分散問題などを解く多くの分散アルゴリズムが提案されているが[(3),(7),(9)-(11)]，それらの多くは，PE やリンクに故障があると正しく動作しない．実際には，PE やリンクに故障が生じることが少なくないので，PE やリンクの故障を考慮した分散アルゴリズムを開発することは重要なことである[(2),(4)-(6),(8)]．

ネットワークでは，PE やリンクの増設や除去が行われることもあるので，分散アルゴリズムでは，なるべくネットワークに関する大域的な情報（ネットワークのトポロジ，PE 数，連結度など）を利用しない方が望ましい．従って，各 PE で利用できるネットワークに関する大域的な情報が分散アルゴリズムにどのような影響を与えるかを明確にすることは興味深い問題である[(5),(6),(10),(11)]．

故障を考慮した分散アルゴリズムを開発するための基礎的な問題の1つとして，PE やリンクの故障を診断する問題がある．文献(6)では，PE の故障を診断する問題を取り上げ，この問題を解く分散アルゴリズムが存在するための条件について考察している．

本稿では，PE やリンクに故障があるネットワークで，任意の1つの PE が自分に接合するリンクのうち任意の1つが故障しているかどうかを診断する問題（リンク故障診断問題）について考察する．特に，以下の要因がこの問題を解く分散アルゴリズムの存在性にどのような影響を与えるかについて考察する．

(1) 各 PE でネットワークに関するどのような大域的な情報が利用できるか．

(2) PE や通信の同期性を仮定するかどうか．

(3) ネットワーク全体での PE やリンクの故障状況（分布）がどのようになっているか．

本稿で得られた結果を表１にまとめておく．

PE の故障診断に関する結果[6]との特徴的な相違点は，次の点である．PE 故障診断の場合，通信が非同期式（L非同期式という．次節参照）なら，PE の故障とその PE に接合するすべてのリンクの通信遅延が大きいこととが区別できないため，PE の故障を診断できない．[6] しかし，リンク故障診断の場合，PE が非同期式（P非同期式という．次節参照）であっても，通信が同期式なら（L同期式という．次節参照），リンクの故障とそのリンクに接続する PE の動作が遅いこととが区別でき，リンク故障診断問題が解けることがある．

2．諸定義

グラフ $G=(V,E)$ は，空でない頂点集合 V と，辺（相異なる頂点 $u, v \in V$ の順序のない対で，これを (u, v) と表す）の集合 E からなる．従って、多重辺も自己ループ（(v,v) の型の辺）も存在しない．$G=(V,E)$ の頂点 $v \in V$ に対し，v の次数 $deg(v)$ を $deg(v)=|\{(v,w) \in E | w \in V\}|$ と定義する．

グラフ $G=(V,E)$ の互いに異なる頂点の系列 $\langle v_1, \ldots, v_m \rangle$ が，各 i $(1 \leqq i \leqq m-1)$ に対し，$(v_i, v_{i+1}) \in E$ を満たすとき，この系列を v_1-v_m 通路とよび，$m-1$ をその通路の長さという．

k を正整数とする．$G=(V,E)$ の $|E'|=k-1$ なる任意の辺集合 $E' \subseteq E$ に対して，$G'=(V,E-E')$ において任意の相異なる２頂点 $u, v \in V$ 間に u-v 通路があるとき，G は k-辺連結であるという．特に，1-辺連結のことを，単に，連結ということがある．

ある２頂点 u, v に対し，長さ２以上の u-v 通路と辺 (u,v) があるとき，閉路が存在するという．閉路のない連結なグラフを木という．根と呼ぶ１頂点を明示した木を根付き木という．u を根とする根付き木において，$\langle v_1(=u), \ldots, v_m \rangle$ が通路のとき，各 i $(1 \leqq i \leqq m-1)$ に対し，v_i を v_{i+1} の親，v_{i+1} を v_i の子といい，各 i, j $(i \leqq j)$ に対し，v_j を v_i の子孫という．また，子のない頂点を葉とよぶ．

$G=(V,E)$ と $G'=(V',E')$ が $V' \subseteq V$ かつ $E' \subseteq E$ を満たすとき，G' を G の部分グラフという．特に，$V'=V$ かつ G' が木のとき，G' を G の生成木という．

［定義１］ネットワークは，２項組 $N=(P,L)$ で定義される．ここで，

１．P はプロセッサ（以下，PE と略記する）の集合．PE は互いに異なる識別子を持ち，PE u の識別子を $id(u)$ と表す．また，$|P|$ を N のサイズという．

２．L は P の相異なる要素の順序のない対（リンクという）の集合．$(u,v) \in L$ のとき，u, v 間に全２重通信路が存在し，u と v の間で，両方向に独立に（反対方向のメッセージに影響を受けることなく）メッセージのやりとりができる．□

ネットワークはグラフとみなせるので，グラフに対する用語や記法をネットワークに対しても用いる．

［定義２］ PE u は，ポートの集合 $PO(u)=\{1, \ldots, deg(u)\}$ を持つ RAM (Random Access Machine)[1] である．各ポートは u に接合する各リンクと１対１に対応する．u のポート a がリンク (u,v) に対応するとき，v を u の a-隣接 PE とよび，この対応関係（ポート対応とよぶ）を $\mu(u,a)=(u,v)$ と表す．リンク (u,v) は，２つのキュー $q(u \to v)$, $q(v \to u)$ からなる．$\mu(u,a)=(u,v)$ のとき，$\mu o(u,a)=q(u \to v)$, $\mu i(u,a)=q(v \to u)$ と表す．□

本稿では，次のことを仮定する．

［仮定１］ ネットワークには共有メモリは存在せず，PE 間の通信はメッセージ交換でのみ行われる．□

［仮定２］（通信の逐次性） リンクを構成するキューはサイズに制限のない FIFO (First-In First-Out) キュー[1]とする．□

［定義３］ PE u のプログラムでは，以下の通信命令を用いることができる．

１．rec(M) (M は変数)：各 a $(a \in PO(u))$ に対し，$\mu i(u,a)$ から先頭のいくつかのメッセージが読みこまれ，a との対にして，M に記憶される．つまり，

$$M := \bigcup_{a \in PO(u)} Ma$$

が実行される．ここで，

$$Ma = \begin{cases} \phi & \text{(rec(M) の実行で } \mu i(u,a) \text{ からメッセージが読みこまれないとき)} \\ \{(a,m_1), \ldots, (a,m_k)\} & \text{(rec(M) の実行で } \mu i(u,a) \text{ からメッセージ } m_1, \ldots, m_k \text{ が読みこまれたとき．但し，集合として得られるので，到着順序についての情報は利用できない)} \end{cases}$$

とする．読みこまれたメッセージはそのキューの先頭から除去される．但し，PE は読みこむメッセージ数を指定できず，いくつのメッセージが読みこまれるかは前もって分らない．特に，$\ell=\mu i(u,a)$ が空でなくても，ℓ からメッセージが読みこまれず，ℓ の内容に変化が生じないことがある．これは，通信遅延があり，それが前もって分らないことを意味する．

２．send(a,m) $(a \in PO(u)$, m は変数)：$\mu o(u,a)$ に

最後の要素として m の内容を加える. □

[仮定3]（通信の無損失性） PE u が rec(M) を繰り返し行えば，任意の a（a∈PO(u)）に対し，μi(u,a) のメッセージはいつか必ず読みこまれる. □

[仮定4] 分散アルゴリズム（以下では，単にアルゴリズムという）は，各 PE が実行するプログラムによって表される．各 PE には，1つのプログラムが搭載されているが，本稿では，全ての PE のプログラムは同一であると仮定する．但し，PE にはネットワークの形状に関するいくつかの定数（形状定数とよぶ）があり，それらのいくつかを利用できるとする．PE u が利用できる可能性のある形状定数として，次の定数を考える．但し，ネットワークを N=(P,L)（$P=\{u_i|1\leqq i\leqq n\}$）とする．

ID_u†：u の識別子 id(u).

DEG_u：u の次数 deg(u).

NL_u：u の隣接 PE の識別子のリスト．$NL_u[a]$（a∈PO(u)）は，u の a-隣接 PE の識別子を表す．

SIZE：ネットワークのサイズ（PE 数）.

CON：CON=k はネットワークが k-辺連結であることを意味する．

MID：各 PE の識別子を表す n×1 行列.

$MID[i] = id(u_i)$ $(1\leqq i\leqq n)$

MPT：PE間の隣接関係とポート対応を表す n×n 行列.

$$MPT[i,j]=\begin{cases} a & (u_j \text{ が } u_i \text{ の a-隣接 PE のとき}) \\ 0 & ((u_i,u_j)\not\in L \text{ のとき}) \end{cases}$$ □

各 PE が初期状態で使える形状定数によって，解ける問題のクラスやアルゴリズムの効率が異なることがある[(5),(6),(10),(11)]．従って，各 PE が初期状態で使える形状定数を明示することは重要である．

[定義4] 形状定数のうち，ID_u，DEG_u を基本情報という．各 PE u が初期状態で利用できる形状定数によって，アルゴリズムを次のように分類する．

基本情報アルゴリズム：基本情報だけを利用できる．

隣接識別子既知アルゴリズム：基本情報と NL_u だけを利用できる．

サイズ既知アルゴリズム：基本情報と SIZE だけを利用できる．

辺連結度既知アルゴリズム：基本情報と CON だけを利用できる．

トポロジ・アルゴリズム：基本情報と MID，MPT だけを利用できる． □

基本情報はどのクラスのアルゴリズムでも利用できるので，基本情報アルゴリズムで解ける問題は，他のどのクラスのアルゴリズムでも解ける．また，MID，MPT からネットワークの形状に関するあらゆる情報が導き出せるので，トポロジ・アルゴリズムで解けない問題は，他のどのクラスのアルゴリズムでも解けない．

本稿では，PE やリンクに故障がある場合について考察する．但し，次のことを仮定する．

[仮定5] PE に故障があると，その PE は一切プログラムを実行しない．従って，故障している PE はメッセージの送信も受信も行わない．PE が故障すると他の PE がその故障を自動的に検知できるというモデルもあるが，ここでは，PE が故障しても他の PE はその故障を直接的には検知できないものとする． □

[仮定6] リンクに故障があると，そのリンクを用いた1方向または2方向の通信ができなくなる（rec 命令を実行してもメッセージが受信できない）．故障しているリンクに send 命令や rec 命令を行うと故障を自動的に検知できるというモデルもあるが，ここでは，故障のあるリンクに send 命令や rec 命令を行っても，その故障は自動的には検知できないものとする． □

以下では，故障している PE，リンクをそれぞれ故障 PE，故障リンクとよび，故障していない PE，リンクをそれぞれ健全 PE，健全リンクとよぶ．リンク $\ell=(u,v)$ に対し，故障のため ℓ を使って u から v にメッセージを送れないとき，$\ell(u\rightarrow v)$ が故障しているという．

[定義5] プログラムで用いられる変数のうち，入力変数，出力変数として着目するものの集合を IV，OV と表す．そして，ネットワーク N=(P,L) について，$IV_N=\{x_u|x\in IV,\ u\in P\}$，$OV_N=\{x_u|x\in OV,\ u\in P\}$ とする．特に入出力命令は考えず，初期状態で入力変数にある値（入力値）が設定されており，終了時に出力変数がある値（出力値）を保持しているものとする．

アルゴリズム実行開始時の IV_N の値（$IV_N{}^0$と表す）が満たすべき関係を表す述語（入力述語）を α_N，アルゴリズム実行終了時の OV_N の全変数の値と IV_N の初期値 $IV_N{}^0$との間で成立すべき入出力対応を与える述語を β_N とする．α_N を満たす任意の $IV_N{}^0$を IV_N の初期値とし，全てのリンクが空である状況で，全ての健全 PE がアルゴリズムAで定められたプログラムの実行を開始するとする．各 PE がプログラムの実行を開始するタイミング，各 PE の動作速度，メッセージの伝搬遅延などの違いにより，様々な実行経過が考えられる．どの実行経過においても，全ての健全 PE は有限個の命令の実行後停止し，全ての健全 PE が停止した時点の OV_N の全変数の値と $IV_N{}^0$との間で β_N が満たされているなら，Aは分散個別問題（以下では，個別問題という）$\pi_N=(\alpha_N,\beta_N)$ を解くという．

ネットワークの任意の集合を $\mathcal{N}$，個別問題の集合を $\pi=\{\pi_N=(\alpha_N,\beta_N)|N\in\mathcal{N}\}$ とする．アルゴリズムAが，全ての $N\in\mathcal{N}$ について π_N を解くとき，Aは π を解く

† 各 PE は同じプログラムを搭載しているので，同じ定数（変数）名を使っている．そこで，PE u の定数（変数）x を x_u と表して区別する．

という．□

初期状態で定数 NL を利用できなくても，PE やリンクに故障がなければ，すべての隣接 PE 間で識別子を交換することができ，NL の内容を知ることができる．また，同様に，その情報を全ての PE に伝えることができ，各 PE で定数 MID，MPT の内容を知ることができる．従って，PE やリンクに故障がなければ，基本情報アルゴリズム，隣接識別子既知アルゴリズム，サイズ既知アルゴリズム，辺連結度既知アルゴリズム，トポロジ・アルゴリズムで解ける問題のクラスに差はない[6]．しかし，PE やリンクに故障があるとき，これらのアルゴリズムで解ける問題のクラスに差がある．本稿では，リンク故障診断問題について，この差を示す．但し，以下では，連結なネットワークについてのみ考察し，連結なネットワーク全ての集合を$\mathcal{N}$と表す．また，PE やリンクの故障について次のように仮定する．

［仮定7］アルゴリズム実行中は PE やリンクの故障が新たに生じたり，故障 PE や故障リンクが復旧したりしない．□

［定義6］ リンク故障診断問題 πl

ネットワーク $N=(P,L)$ において，入力変数 $test_u$ の初期値が true である PE u $(\in P)$ が正確に１個だけ存在し（この PE を診断 PE という），u の入力変数 $link_u$ の初期値が a $(\in PO(u))$ のとき，u がリンク $\mu(u,a)$ が故障しているかどうかを判定する問題である．つまり，$\pi l=\{\pi l_N=(\alpha l_N,\beta l_N) \mid N=(P,L)\in\mathcal{N}\}$ は，次のように定義される．

各 PE の入力変数：test（論理型），link（整数型）．

各 PE の出力変数：fail（論理型）．

αl_N：ある $u\in P$ の入力変数 test，link は，

$test_u$ = true，かつ，$link_u \in PO(u)$

を満たす．また，任意の $v\in P-\{u\}$ に対し，

$test_v$ = false

が，成り立つ．

βl_N：診断 PE を u とし，初期状態で $link_u=a$ $(\in PO(u))$ とする．u が健全 PE ならば，u の出力変数 $fail_u$ は，次のように定まる．

$$fail_u=\begin{cases}\text{true} & (\mu(u,a)\text{ が故障リンクのとき})\\ \text{false} & (\mu(u,a)\text{ が健全リンクのとき})\end{cases}$$ □

［定義7］PE やリンクに故障のあるネットワークにおいて，u_1-u_m 通路 $\langle u_1,u_2,\ldots,u_m\rangle$ が次の２条件を満たすとき，この通路は健全であるという．

(1) 各 i $(1\leqq i\leqq m)$ に対し，u_i は健全 PE．

(2) 各 i $(1\leqq i<m)$ に対し，(u_i,u_{i+1}) は健全リンク．□

以下では，同期に関する定義を行う．まず，アルゴリズムの任意の実行（経過）に対し，通信順序とよぶ系列を定義する．これは，アルゴリズム実行時における通信命令の実行順序を表すものである．次の定義では，グローバル時刻を想定し，グローバル時刻によって，通信命令を実行した順序を定めている．但し，グローバル時刻は通信順序を定義するためにだけ用いられるものであり，PE がアルゴリズム実行時にグローバル時刻を利用したりはできない．

［定義8］通信順序は PE の集合の有限または無限系列であり，アルゴリズムの任意の実行に対し定義され，通信命令を実行した PE の順序を表す．つまり，通信順序 $S=(s_1,s_2,\ldots)$ は，アルゴリズム実行時に最初に通信命令を実行したのは s_1 に属する各 PE であり，その次に通信命令を実行したのは s_2 に属する各 PE である（以下同様）ことを表す．但し，通信命令の実行にかかる時間は考えない．□

［例１］あるアルゴリズムを $N=(P,L)$（但し，$P=\{v_1, v_2, v_3\}$）で実行したときの通信命令の実行の様子を図１に示す．このときの通信順序 S は次のようになる．

$S=(s_1=\{v_1,v_2\},\ s_2=\{v_2\},\ s_3=\{v_1,v_2,v_3\},\ s_4=\{v_2\},\ s_5=\{v_3\},\ s_6=\{v_1,v_3\},\ s_7=\{v_3\},\ s_8=\{v_1,v_2\},\ s_9=\{v_1\})$ □

通信順序を $S=(s_1,s_2,\ldots)$ とする．各 PE は逐次的にプログラムを実行するので，PE u が通信命令を q 回実行するなら，$u\in s_i$ なる i は正確に q 個存在する．S に対し，S の部分系列 $(s_p,\ldots,s_q)$ $(1\leqq p\leqq q)$ を $S[p,q]$ と表す．そして，$S'=(s'_1,s'_2,\ldots)$ を通信順序の部分系列とするとき，PE u に対し，$u\in s'_i$ なる i が q 個存在するとき，u が S' に q 回現れるという．また，S'に u が現れる回数を $\#(u,S')$ と表す．

［定義9］ある定数$\Gamma(\geqq 1)$ が存在し，ネットワーク $N=(P,L)$ における任意のアルゴリズムの任意の通信順序 $S=(s_1,s_2,\ldots)$ が次の条件（PS）を満たすとき，N はＰ同期式であるといい，ΓをＰ同期定数とよぶ．

$$\begin{aligned}(\mathrm{PS})\ &\forall u(\in P)\forall q\forall q'(1\leqq q\leqq q')\\ &\quad[\exists v(\in P)[\#(v,S[q,q'])\geqq\Gamma]\\ &\qquad\wedge\exists q''(q'<q'')[u\in s_{q''}]\\ &\qquad\qquad\Rightarrow \#(u,S[q,q'])\geqq 1]\end{aligned}$$

N がＰ同期式でないとき，Ｐ非同期式という．□

Ｐ同期式ネットワーク（ΓをＰ同期定数とする）で

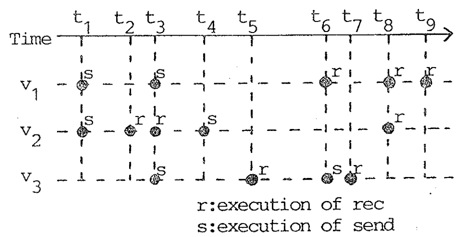


図１ $N=(\{v_1,v_2,v_3\},L)$ での通信命令の実行例

表1　リンク故障診断問題（PE v によるリンク $\ell=(v,w)$ の故障診断）を解くアルゴリズム

<table>
<tr><th>アルゴリズム
ネットワーク</th><th>基本情報
アルゴリズム</th><th>サイズ既知
アルゴリズム</th><th>辺連結度既知
アルゴリズム</th><th>隣接識別子既知
アルゴリズム</th><th>トポロジ・
アルゴリズム</th></tr>
<tr><td>L非同期式</td><td colspan="5">ℓ以外のリンクや PE に故障がなくても存在しない（定理1）</td></tr>
<tr><td>PL同期式</td><td colspan="5">PLSB1（故障 PE 数，故障リンク数にかかわらず解く）（定理2）</td></tr>
<tr><td rowspan="4">L同期式かつ
P非同期式</td><td rowspan="4">ℓ以外のリンクや PE に故障がなくても存在しない
（定理7(3)）</td><td colspan="2">v，w 以外の任意のある PE が故障していると存在しない（定理5）</td><td colspan="2" rowspan="4">LSN1
（故障 PE 数，故障リンク数にかかわらず解く）
（定理4）</td></tr>
<tr><td colspan="2">PE に故障がないとき</td></tr>
<tr><td rowspan="2">LSS1（故障リンク数にかかわらず解く）
（定理6）</td><td>LSC1（故障リンク数が $\lceil CON/2 \rceil -1$ 以下のとき解く）
（定理7(1)）</td></tr>
<tr><td>故障リンク数が $\lceil CON/2 \rceil$ 以上のとき存在しない
（定理7(2)）</td></tr>
</table>

は，任意のある PE が Γ 回通信命令を実行する間に通信命令を実行しない PE は，それ以後も通信命令を実行しない．このことは，PE の実行時間は通信命令の実行回数に比例するという仮定のもとで，各 PE の動作速度の比の上界が Γ であることを意味する．

［定義10］ある定数 $\Delta(\geq 1)$ が存在し，ネットワーク N における任意のアルゴリズムの任意の実行の通信順序 $S=(s_1,s_2,...)$ が次の条件（LS）を満たすとき，N はL同期式であるといい，Δ をL同期定数とよぶ．

（LS）q，$q'(q'>q\geq 1)$ を任意の自然数とする．$u\in s_q$ で，このとき（s_q に対応する通信命令として）u は send 命令を実行し，隣接 PE v にメッセージ mes を送信したとする．そして，リンク（u,v）は故障していないとする．ある PE w が存在して，$\#(w,S[q+1,q'])\geq\Delta$ かつ $v\in s_{q'}$ かつこのとき v が rec 命令を実行したなら，ある q''（$q+1\leq q''\leq q'$）が存在して，$v\in s_{q''}$ かつこのとき v が rec 命令を実行し，メッセージ mes を受信している．

N がL同期式でないとき，L非同期式という．□

L同期式ネットワーク（Δ をL同期定数とする）では，u が v に送信したメッセージは，任意のある PE が Δ 回通信命令を実行してから v が rec 命令を実行すれば，遅くてもそのとき受信される．つまり，PE の実行時間が通信命令の実行回数に比例するという仮定のもとで，メッセージがリンクを伝わる伝搬遅延時間と PEの動作速度の比の上界が Δ であることを意味する．

ネットワークが，P同期式かつL同期式のとき，PL同期式であるという．また，P非同期式かつL非同期式のとき，非同期式であるという．

各 PE が利用できる定数として，次のものを考える．

［定義11］同期に関する定数（同期定数とよぶ）で，PE が利用できる定数として PSYNC，LSYNC があり，それぞれP同期定数，L同期定数を表す．□

但し，同期定数について，次の仮定を置く．

［仮定8］ネットワークがP同期式（L同期式）なら，各 PE は初期状態で同期定数 PSYNC（LSYNC）を利用できる．□

3．リンク故障診断問題

リンク故障診断問題 $\pi 1$ について考察する．但し，診断 PE を v，$link_v=a$，$\mu(v,a)=(v,w)$ とする．つまり，v によるリンク $\ell=(v,w)$ の故障診断を考える．診断 PE v が故障していると $\pi 1$ の意味がなくなる．また，PE w が故障していると，ℓ が故障しているかどうかにかかわらず，ℓ を用いる通信ができないので，$\pi 1$ が解けなくなる．また，ある健全 PE u に対し，健全な v-u 通路が存在しないと，u がいつ停止すればよいかの決定が難しくなる．（PL同期式でないときは決定できないことが示せる．）そして，ℓ が故障しているかどうかを v が判定できても，u が停止しないために，定義上 $\pi 1$を解いたことにならない（定義5参照）．そこで以下では，次の2条件が成立するときに $\pi 1$ が解けるかどうかについて議論する．そして，以下の定理や補題では，特に断らないが，次の2条件を前提としている．

［条件1］PE v，w はともに故障していない．□

［条件2］任意の健全 PE u に対して，健全な v-u 通路が存在する．□

$\pi 1$ を解くアルゴリズムを示した場合，条件1，2が成り立たないときに，そのアルゴリズムがどのような動作をするかを（注）に示す．（注）より，本稿のアルゴリズムはすべて，条件1，2が成り立たないときには，停止しないか，あるいは ℓ が故障していると診断することがわかる．従って，条件1，2が成立しなくても，本稿のアルゴリズムが ℓ を健全リンクと診断したときにはその結果は正しい．

表1に，本稿の結果をまとめておく．

3．1　L非同期式ネットワーク

次の定理より，L非同期式の場合，$\pi 1$が解けない．

［定理1］L非同期式のとき，P同期式で ℓ 以外のリ

ンクや PE に故障がなくても，$\pi 1$ を解くトポロジ・アルゴリズムは存在しない．

（略証）L非同期式ネットワークでは，ℓが故障しているのか，ℓの伝搬遅延が大きいのかが有限時間内に区別できない．　□

３．２　ＰＬ同期式ネットワーク

ＰＬ同期式ネットワークで$\pi 1$ を解く基本情報アルゴリズム PLSB1 を示す．まず，PLSB1 の基本方針とそこで使われるメッセージについて，簡単に説明する．このアルゴリズムは，次の３段階からなる．

(1) メッセージ《T1》を vから w へℓを通じて送信する．これを w が受信すると，$\ell(v \to w)$ が故障していないことがわかる．

(2) (1) で w が《T1》を受信したとき，メッセージ《T2》を w から v へℓを通じて送信する．これを v が受信すると，ℓが故障していないことがわかる．

(3) ℓが故障しているかどうかを v がわかると，v は全ての PE を停止させる．このためにメッセージ《HLT》を用いる．

（注）以降で述べるアルゴリズムでも同じ基本方針を用い，《T1》,《T2》,《HLT》を同じ目的に用いる．

アルゴリズム PLSB1 を以下に示す．但し，B がポート集合を表すとき，B の各ポートに対して任意の順でメッセージ mes を送信することを send(B,mes) と略記する．また，$PO_u = \{1,2,...,DEG_u\}$ とする．

［アルゴリズム PLSB1］

(PL-Synchronized algorithm with Basic-information for $\pi 1$)

【診断 PE v の動作】†

send($link_v$,《T1》) を実行し，その後，rec 命令を 2×(PSYNC+LSYNC)-1 回実行する．この間に《T2》を受信すればℓは健全リンクであると診断し，《T2》を受信しなければℓは故障リンクであると診断する．そして，send(PO_v,《HLT》) を実行し，停止する．

【v 以外の各 PE u の動作】

《HLT》を受信するまで，rec 命令を繰り返し実行する．この間に，《T1》を受信すると，send(b,《T2》)(b は《T1》を受信したポート) を実行する．《HLT》を受信すると，send(PO_u-C,《HLT》) (C は《HLT》を受信したポートの集合) を実行し，停止する．　□

PLSB1 は，文献(6)のアルゴリズム PLSBpとメッセージが異なる(PLSB1 の《T1》,《T2》はそれぞれ，PLSBp の《TEST》,《SAFE》に対応) だけで，他は同じである．文献(6)と同様にして，次の定理が証明できる．

†全ての PE のプログラムは同一であると仮定したが，ここでは，v と v 以外の PE とが実際に実行する部分だけを示す．実際のプログラムでは，入力変数 test の値によってどちらか一方に分岐する．

［定理２］ＰＬ同期式のとき，基本情報アルゴリズム PLSB1 は，故障 PE 数，故障リンク数にかかわらず$\pi 1$ を解く．　□

（注2.1）PE w が故障 PE のとき（条件１が成立しないとき），v は《T2》を受信しないので，ℓが故障していなくてもℓは故障していると診断する．

（注2.2）条件２が成立しなくても，v はℓが故障しているかどうか判定できる（v の停止時，ℓが故障リンクなら $fail_v$=true，健全リンクなら$fail_v$=false が成り立つ）が，v との間に健全な通路のないPE は《HLT》が届かないので，停止しない．v 以外の PE の動作を，PSYNC+LSYNC 回 rec 命令を実行してもメッセージを受信しなければ停止するように変更すれば，条件２が成立しなくても全ての PEが有限個の命令の実行後停止するので，同様のアルゴリズムで$\pi 1$ が解ける．

３．３　Ｌ同期式かつＰ非同期式ネットワーク

3.3.1 健全な v-w 通路の必要性

条件１，２より，健全な v-w 通路が存在する．条件２は，全ての PE を停止させるために設けた条件である．しかし，次の定理より，Ｌ同期式かつＰ非同期式の場合は，健全な v-w 通路が存在することは，w を停止させるためだけに必要な条件ではなく，v がℓの故障を判定するために必要な条件である．

［定理３］Ｐ非同期式のとき，健全な v-w 通路が存在しなければ，Ｌ同期式で故障 PE が存在しなくても，v は$\ell=(v,w)$ の故障を判定できない．つまり，v の停止時（v 以外の PE は停止しなくてもよい）に，ℓが故障リンクなら $fail_v$=true，ℓが健全リンクなら $fail_v$=false となるトポロジ・アルゴリズムは存在しない．

（証明）Ｐ非同期式ネットワークで，健全な v-w 通路がなくても，v がℓの故障を判定できるトポロジ・アルゴリズムＡが存在すると仮定する．

w が v だけに隣接しているようなネットワーク N で$\ell=(v,w)$ だけが故障していて他に故障がないときＡを実行すると，Ａの仮定より，$fail_v$=true で vは停止する．但し，ℓは両方向とも故障している場合を考える．このとき，v はℓからメッセージを受信しない．

一方，どの PE もリンクも故障していない N でＡを実行するとする．N がＰ非同期式なので w の動作が非常に遅い場合がある．このとき，v がℓからメッセージを受信する前に，w 以外の各 PE は，ℓだけが両方向とも故障しているときと同じ動作（各 PE は同じ時刻に同じ命令を実行する．さらに，その命令が rec 命令なら，同じポートから同じメッセージを受信する）をすることがある．すなわち，ℓが故障していないにもかかわらず，vが $fail_v$=true で停止する．このことは，v がℓの故障を判定できることに矛盾する．　□

定理３より，Ｐ非同期式ネットワークにおいては，

健全な v-w 通路がなければ，v は ℓ が故障しているかどうかを判定できない．一方，v，w 以外のある健全PE u に対し，健全な v-u 通路がなくても，v は ℓ が故障しているかどうかを判定できることがある．(但し，u は停止しないかもしれない．) そこで，条件２を次のように２つの条件に分けることがある．以下でも，これまで通り，条件２が成り立つ場合について議論する．

［条件2.1］健全な v-w 通路が存在する．□

［条件2.2］v，w 以外の任意の健全 PE u に対して，健全な v-u 通路が存在する．□

文献(6)は，Ｌ非同期式ネットワークでは，PE 故障診断問題が解けないことを示した．これは，PE の故障と，その PE に接合する全てのリンクの伝搬遅延が大きいこととが区別できないためである．しかし，リンク故障診断の場合は，PE が利用できる形状定数によっては，Ｐ非同期式であっても，リンクの故障と，そのリンクに接続する PE の動作が遅いこととが区別できる場合があることを以下に示す．

3.3.2 隣接識別子既知アルゴリズム

まず，故障 PE 数，故障リンク数にかかわらず $\pi 1$ を解く隣接識別子既知アルゴリズム LSN1 を示す．ここでは，《T1》，《T2》，《HLT》以外に，以下のメッセージを用いる．

《S1,id(w)》：id(w) は w の識別子で，$NL_v[link_v]$ の値．v が《T1》の送信後に送信する．ℓ 以外の健全な v-w 通路を通って w に伝えられる．w がこれを受信すると，w は自分に接合するリンクの故障診断が行われていることが分る．

《S2》：w が《T2》の送信後に送信する．ℓ 以外の健全な w-v 通路を通って v に伝えられる．v がこれを受信すると，vは ℓ に《T2》が送信されたことが分る．

《FAIL》：w が $\ell(v \to w)$ が故障していると分ったときに，そのことを v に知らせるために使う．健全な w-v 通路を通って v に伝えられる．

［アルゴリズム LSN1］

(L-Synchronized algorithm with Neighbors' identities for $\pi 1$)

(1) 診断 PE v は，send($link_v$，《T1》) を実行後，send(PO_v-{$link_v$}，《S1,id(w)》) を実行する．《S1,id(w)》は放送（これを初めて受信した PE は，それを受信した以外の全てのリンクに送信する）によって，PE w に伝えられる．

(2) w は，《T1》を受信する（$\ell(v \to w)$ が故障していないことがわかる）と，send(b，《T2》)（b は《T1》を受信したポート）を実行後，send(PO_w-{b}，《S2》) を実行する．《S2》は放送によって，v に伝えられる．《T1》受信前に，w が《S1,id(w)》を受信した場合，w はさらに rec 命令を LSYNC-1 回実行する．この間に《T1》を受信すると，上記と同様，《T2》，《S2》を送信する．この間に《T1》を受信しない（$\ell(v \to w)$ が故障しているとわかる）場合は，《FAIL》を放送によって v に伝える．

(3) v は，《T2》を受信すると，$fail_v$:=false（ℓ を健全リンクと診断）を実行し，(4)へ．《FAIL》を受信すると，$fail_v$:=true（ℓ を故障リンクと診断）を実行し，(4)へ．《T2》受信前に，v が《S2》を受信した場合，v はさらに rec 命令を LSYNC-1 回実行し，この間に《T2》を受信すると，$fail_v$:=false を実行し，(4)へ．この間に《T2》を受信しない場合は，$fail_v$:= true を実行し，(4)へ．

(4) v は，全ての隣接 PE に《HLT》を送信し，停止する．《HLT》は放送によって，全ての健全 PE に伝えられ，これを受信した PE は停止する．□

［定理４］Ｌ同期式のとき，隣接識別子既知アルゴリズム LSN1 は，故障 PE 数，故障リンク数にかかわらず $\pi 1$ を解く．

（証明）(a) ℓ が健全リンクのとき，LSN1 は必ず停止し，停止時に $fail_v$=false が成立することを示す．

w が《S1,id(w)》受信前に《T1》を受信すれば，w は《T2》をリンク ℓ を通じて v に送信する．w が《T1》受信前に《S1,id(w)》を受信したとする．w が受信した《S1,id(w)》は v が送信したものが中継されてきたものである．v が《T1》送信後に《S1,id(w)》を送信すること，及び，Ｌ同期式であることより，w が《S1,id(w)》受信後に rec 命令を LSYNC-1回実行すれば，この間に《T1》を受信する．そして，wは(2)で《T2》をリンク ℓ を通じて v に送信する．同様にして，v が《T2》を受信することを示せる．

v が《T2》を受信すると，v は $fail_v$:=false を実行後，《HLT》を送信して停止する．条件２より，任意の健全 PE u に対し健全な v-u 通路が存在するので，すべての PE は《HLT》を受信し，停止する．従って，LSN1 が停止する．

(b) ℓ が故障リンクのとき，LSN1 は必ず停止し，停止時に $fail_v$=true が成立することを示す．但し，v が停止すれば LSN1 が停止することを(a)と同様にして示せるので，ここでは，$fail_v$=true で v が停止することを示す．

(b1) $\ell(v \to w)$ が故障している場合．

条件2.1より，健全な v-w 通路が存在するので，w は《S1,id(w)》を受信し，LSYNC-1 回 rec 命令を実行する．$\ell(v \to w)$ が故障しているので，w はこの間に《T1》を受信せず，全ての隣接 PE に《FAIL》を送信する．同様に，v は《FAIL》を受信し，$fail_v$:=true を実行し，停止する．

(b2) $\ell(v \to w)$ は故障していないが，$\ell(w \to v)$ が故

障している場合．

(a)の場合と同様に，w は《T1》を受信する．そして，w は《T2》，《S2》を送信する．条件2.1より，健全な v-w 通路が存在するので，v は《S2》を受信する．v は《S2》受信後，rec 命令を LSYNC-1 回実行するが，$\ell(w \to v)$ が故障しているので，《T2》を受信しない．従って，v は $fail_v$:=true を実行し，停止する． □

(注4.1) w が故障 PE のとき，vは《T2》も《FAIL》も受信しないので，LSN1 は停止しない．

(注4.2) 条件2.2が成立しなくても，LSN1 で v は ℓ が故障しているかどうか判定できる（(注2.2) 参照）が，v 以外の PE で停止しないものが存在する．

3.3.3 故障 PE がある場合のサイズ既知，辺連結度既知アルゴリズム

［定理５］ L同期式かつP非同期式で，v，w 以外の PE が故障している可能性があるとき，v，w 以外の故障 PE が高々１個で，ℓ 以外のリンクに故障がなくても，次のことが成り立つ．

(1) $\pi 1$を解くサイズ既知アルゴリズムは存在しない．

(2) $\pi 1$を解く辺連結度既知アルゴリズムは存在しない．

(証明) (1)の証明：ℓ 以外のリンクに故障がなく，v，w 以外の故障 PE が高々１個のときに$\pi 1$を解くサイズ既知アルゴリズムAが存在すると仮定する．Aは，ネットワーク N=(P,L) の ℓ，w' (v，w 以外の任意の PE) だけが故障しているとき，$\pi 1$ を解き，ℓ を故障リンクと診断して停止する．但し，ℓ は両方向とも通信できない場合を考える．また，N において，v は w の b-隣接 PE とする．

N'=(P',L') を次のように定義する．(図２)

$$P' = P,\ L' = (L-\{(v,w)\}) \cup \{(v,w'),(w,w')\}$$

Aがサイズ既知アルゴリズムであることより，w' 以外の各 PE が N' で利用できる形状定数の値は，N の

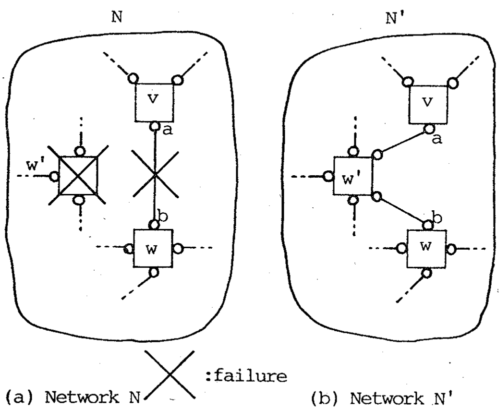


図2 定理5の証明のネットワーク N と N'

場合と同じである（w' は次数が異なる）．また，各 PE u，各 c（$c \in PO_{N'}(u)$†）に対し，N' のポート対応を次のように定める．

$$\mu_{N'}(u,c) = \begin{cases} (v,w') & (u=v,\ c=a(=link_v) \text{ のとき}) \\ (w,w') & (u=w,\ c=b \text{ のとき}) \\ (w',v) & (u=w',\ c=deg_{N'}(w')-1 \text{ のとき}) \\ (w',w) & (u=w',\ c=deg_{N'}(w') \text{ のとき}) \\ \mu_N(u,c) & (\text{その他の場合}) \end{cases}$$

どの PE もリンクも故障していない N' で，v が診断 PE，$link_v$=a のときにAを用いて$\pi 1$（v が (v,w') の故障を診断する問題）を解くことを考える．P非同期式なので，w' の動作が非常に遅い場合がある．このとき，w' 以外の PE は，ℓ，w' だけが故障している N で$\pi 1$ を解いたときと同じ動作（定理３の証明参照）をする場合がある．このとき，(v,w') を故障リンクと診断して停止する．このことは，Aが$\pi 1$ を解くことに矛盾する．

(2) の証明：N が k-辺連結なら(1)で作った N' もk-辺連結であることが示せる．辺連結度既知アルゴリズムであることより，w' 以外の各 PE が N' で利用できる形状定数の値は，N の場合と同じである．このことから，(1)と同様にして証明できる． □

3.3.4 故障 PE がない場合のサイズ既知，辺連結度既知アルゴリズム

ここでは，故障 PE がないとき，$\pi 1$ を解くアルゴリズムについて考える．文献(5)のアルゴリズム SST（故障 PE がない非同期式ネットワーク N で，故障リンク数にかかわらず N の生成木を構成するサイズ既知アルゴリズム)，CST（同様のネットワークで，故障リンク数が $\lceil CON/2 \rceil -1$‡ 以下のとき N の生成木を構成する辺連結度既知アルゴリズム）を利用して$\pi 1$ を解くアルゴリズム LSS1 と LSC1 を示す．但し，SST, CST によって構成される生成木には，故障リンクは含まれない[5]．SST, CST は，同期定数 LSYNC を利用しないが，L同期式かつP非同期式の場合にも適用できる．

［アルゴリズム LSS1］

(L-Synchronized algorithm with Size for π 1)

(1) SST を用いてネットワークの生成木 T を構成する．このとき，各 PE u は，変数 TPO_u に T での隣接 PE に通じるポートの集合を求めるものとする．

(2) $link_v \in TPO_v$ なら，v は $fail_v$:=false を実行し，(4)へ．$link_v \notin TPO_v$ なら，v は send($link_v$, 《T1》) を実行する．そして，ダミーの send 命令（例えば，send($link_v$, 《T1》)) を LSYNC-1 回実行してから，send(TPO_v, 《S1》) を実行する．《S1》は生成木 T

† $PO_N(u)$ は，N における u のポート集合を表す．$\mu_N(u,c)$ についても同様．

‡ $\lceil x \rceil$ は x より小さくない最小の整数を表す．

を利用して，全ての PE に伝えられる．各 PE u は，《S1》を受信したとき，それ以前に《T1》を受信していれば rs_u:=true（rs_uは u の論理型局所変数）を実行し，send(b,《T2》)（b は《T1》を受信したポート）を実行する．《T1》を受信していなければ，rs_u:=false を実行する．

(3) T を v を根とする根付き木とみなし，T の葉から順に，各 PE u は，T における u の子孫（u も含む）の rs の論理和を求め，その結果を u の親に送信する．こうして，v で $\bigvee_{u \in V-\{v\}} rs_u$（=RS とする）を求める．RS=false なら，$fail_v$:=true を実行し，(4)へ．RS=true なら，v は rec 命令を LSYNC-1 回実行する．この間，あるいは以前に《T2》を受信すれば，$fail_v$:=false を実行し，(4)へ．《T2》を受信しなければ，$fail_v$:=true を実行し，(4)へ．

(4) T を用いて，v から全ての PE に《HLT》を伝え，アルゴリズムを停止させる． □

［定理６］ L同期式で，故障 PE がないとき，故障リンク数にかかわらず，サイズ既知アルゴリズム LSS1 は$\pi 1$ を解く．

（証明）LSS1 が停止するのは明らか．

(a) ℓが故障していないとき，v は $fail_v$=false で停止することを示す．SST で構成される生成木を T=(P, L') とする．

(a1) $\ell \in L'$ のとき：$link_v \in TPO_v$ であり，v は $fail_v$:=false を実行し，停止する．

(a2) $\ell \notin L'$ のとき：v は，ℓに《T1》を送信後，LSYNC-1 回ダミーの送信命令を実行してから《S1》を放送する．L同期式なので，w は《S1》を受信するまでに《T1》を受信する．従って，rs_w=true となり，RS=true となる．w は子孫の rs の論理和を親に送信する前に，ℓに《T2》を送信する．v は RS を求めた後，rec 命令を LSYNC-1 回実行する．L同期式であることより，遅くともこの間に，v は 《T2》を受信する．従って，v は $fail_v$:=false を実行し，停止する．

(b) ℓが故障しているとき，v は $fail_v$=true で停止することを示す．

(b1) $\ell(v \to w)$ が故障している場合．

《T1》を受信する PE は存在しない．従って，RS=false となり，v は $fail_v$:=true を実行し，停止する．

(b2) $\ell(v \to w)$ は故障していないが，$\ell(w \to v)$ が故障している場合．

RS=true となるが，v は《T2》を受信することはないので，$fail_v$=true で停止する． □

（注）故障 PE が存在するとき，または，条件2.2が満たされないとき，アルゴリズム SST が停止しないので，アルゴリズム LSS1 は停止しない．

次に，辺連結度既知アルゴリズムについて考える．

［アルゴリズム LSC1］

(L-Synchronized algorithm with Connectivity for $\pi 1$)

まず，CST を用いてネットワークの生成木を構成する．その後の動作は LSS1 と同じ． □

［定理７］ リンク故障診断問題$\pi 1$ について，L同期式かつP非同期式で，故障 PE がないとき，次のことが成り立つ．

(1) 故障リンクの数が ⌈CON/2⌉-1 以下のとき，辺連結度既知アルゴリズム LSC1 は$\pi 1$ を解く．

(2) 故障リンクの数が ⌈CON/2⌉ 以上のとき，$\pi 1$ を解く辺連結度既知アルゴリズムは存在しない．

(3) ℓ以外のリンクに故障がなくても，$\pi 1$ を解く基本情報アルゴリズムは存在しない．

（証明）(1)の証明：CST は，故障リンク数が ⌈CON/2⌉-1 以下のとき，生成木を構成する．定理６と同様にして，証明できる．

(2)の証明：故障リンク数が ⌈CON/2⌉ でも，$\pi 1$ を解く辺連結度既知アルゴリズムＡが存在すると仮定する．

ネットワーク N=(P,L) を次のように定義する．

$P = \{v_0(=v), v_1(=w), \ldots, v_{n-1}\}$

$L = \{(v_i, v_j) \mid i \neq j\}$

つまり，N は n 個の PE からなり，任意の相異なる２つの PE 間にリンクがあるネットワークである．このとき，N は (n-1)-辺連結である．

$Lf = \{(v_{2i}, v_{2i+1}) \mid 0 \leq i \leq \lceil (n-1)/2 \rceil - 1\}$

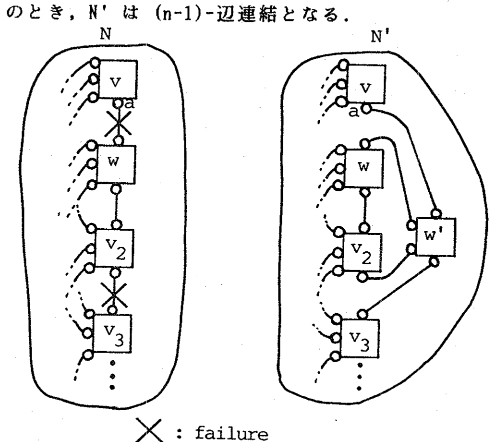
とする．Ａは，CON=n-1 のとき，Lf のリンクだけが全て両方向とも故障していて，他に故障がない N で$\pi 1$ を解き，$\ell(\in Lf)$ が故障していると診断する．

N'=(P',L') を次のように定義する．(図３)

$P' = P \cup \{w'\}$

$L' = (L - Lf) \cup \{(v_i, w') \mid 0 \leq i \leq 2 \times \lceil (n-1)/2 \rceil - 1\}$

このとき，N' は (n-1)-辺連結となる．



(a) Network N　(b) Network N'

図３　定理７の証明のネットワーク N と N'

また，ポート対応を次のように定める．

各 i, b ($0 \leqq i \leqq n-1$, $1 \leqq b \leqq n-1$) に対して，

$$\mu_N'(v_i,b) = \begin{cases} (v_i,w') & (\mu_N(v_i,b) \in Lf \text{ のとき}) \\ \mu_N(v_i,b) & (\mu_N(v_i,b) \notin Lf \text{ のとき}) \end{cases}$$

各 b ($1 \leqq b \leqq n$) に対し，

$$\mu_N'(w',b) = (w',v_{b-1})$$

PE もリンクも故障していない N' で，Aを用いて v が (v,w') を故障診断する場合を考える．ネットワークがP非同期式なので，w' の動作が非常に遅い場合があり，w' 以外の PE は，Lf のリンクだけが全て両方向とも故障している N で$\pi 1$ を解くときと同じ動作（定理３の証明参照）をする場合がある．このとき，(v, w') が故障していないにもかかわらず，故障していると診断する．このことは，Aが$\pi 1$ を解くことに矛盾する．

(3)の証明：ネットワーク N=(P,L) を (2)の証明で用いた N とする．N'=(P',L') を次のように定義する．

$$P' = P \cup \{w'\}$$

$$L' = (L-\{(v,w)\}) \cup \{(v,w'),(w,w')\}$$

このとき，(2) と同様にして，(3) が成立することが示せる． □

（注7.1）故障 PE が存在するとき，または故障リンク数が $\lceil CON/2 \rceil$ 以上のとき，CST が停止しないので，アルゴリズム LSC1 は停止しない．

（注7.2）故障 PE が存在しなければ，故障リンク数が $\lceil CON/2 \rceil -1$ 以下のとき，ネットワークが CON-辺連結であることより，条件2.2は成り立つ．

4．むすび

PE で利用できる大域的情報やネットワークの同期性がリンク故障診断問題を解く分散アルゴリズムに与える影響について考察した．本稿では，リンク故障診断問題が解けるかどうかを議論の中心にしたので，アルゴリズムの効率をよくすることについては考えていない．リンク故障診断問題を解く効率のよいアルゴリズムを考えることは，今後の課題である．

謝辞

本研究は一部文部省科学研究費の補助のもとで行った．

文　献

(1) A.V.エイホ，J.E.ホップクロフト，J.D.ウルマン："アルゴリズムの設計と解析Ｉ"，野崎，野下訳，サイエンス社 (1977)．

(2) D.Dolev, C.Dwork and L.Stockmeyer:"On the minimal synchronism needed for distributed consensus", Proc. of 24th FOCS, pp.393-402 (1983).

(3) R.G.Gallager, P.A.Humblet and P.M.Spira:"A distributed algorithm for minimum-weight spanning trees", ACM TOPLAS, 5, 1, pp.66-77 (1983).

(4) O.Goldreich and L.Shrira:"Electing a Leader in a Ring with Link Failures", Acta Informatica, 24, pp.79-91 (1987).

(5) 増澤，萩原，都倉:"辺故障を考慮したある分散アルゴリズム"，信学論(D)，J69-D，10，pp.1394-1405 (1986)．

(6) 増澤，萩原，都倉:"プロセッサ故障診断のための分散アルゴリズム"，信学論(D)，J70-D，6，pp.1092-1103 (1987)．

(7) 朴，増澤，萩原，都倉:"幅優先生成木構成の分散アルゴリズムについて"，信学技報，COMP86-44，(1986)．

(8) M.Pease, R.Shostak and L.Lamport:"Reaching agreement in the presence of faults", JACM, 27, 2, pp.228-234 (1980).

(9) D.Rotem, K.Korach and N.Santoro:"Analysis of a distributed algorithm for extrema finding in a ring", Tech. Rep. of School of Computer Sci., Carleton Univ., CANADA, SCS-TR-61 (1984).

(10) N.Santoro:"On the message complexity of distributed problems", Int. J. Computer and Information Sciences, 13, 3, pp.131-147 (1984).

(11) N.Santoro:"Sense of direction, topological awareness and communication complexity", SIGACT NEWS, 16, 2, pp.50-56 (1984).