論　文

非停止永久故障に耐性を有する自己安定生成木構成プロトコル

浮穴　学慈[†*]　　片山　喜章[††**]　　増澤　利光[†††]　　藤原　秀雄[†]

A Self-Stabilizing Spanning Tree Protocol that Tolerates Non-Quiescent Permanent Faults

Satoshige UKENA[†*], Yoshiaki KATAYAMA[††**], Toshimitsu MASUZAWA[†††], and Hideo FUJIWARA[†]

あらまし　ネットワークで相互接続された多数のプロセスから構成される分散システムにおいて，プロセス故障にかかわらず正しく動作するプロトコルを設計開発することが重要である．このような故障耐性を有するプロトコルの設計パラダイムとして，自己安定プロトコルが有望視されている．自己安定プロトコルとは，任意のネットワーク状況から実行を開始しても，解を求めて安定するプロトコルである．この性質から，自己安定プロトコルは任意の一時故障に対する故障耐性を有する．本論文では，状態通信モデル上で，一時故障だけでなく，永久故障に対する故障耐性も有する自己安定プロトコルについて考察する．まず，永久故障の新たなクラスとして，非停止永久故障を定義する．そして，ネットワークの生成木を構成する問題に対し，非停止永久故障プロセスが存在しても，問題を解く自己安定プロトコルを提案する．提案するプロトコルの安定時間はたかだか $n(n/2+\mathcal{F})d$ ラウンドである．ここで，n はプロセス数，d はネットワークの最大次数，$\mathcal{F}$ は故障プロセスの状態変化が観測されるまでの時間の上界（定数）である．

キーワード　分散プロトコル，自己安定，故障耐性，非停止永久故障，一時故障，生成木

1. まえがき

ネットワーク環境の発達にともない，ネットワーク上の計算機が協調して問題解決するための計算手続き「分散プロトコル」の重要性が増している．ネットワーク環境（以下，分散システム）の利点として，一部の計算機で障害が起きても，残りの正常な部分でサービスを継続できる可用性（availability）が挙げられる．分散システムの可用性を十分に引き出すためには，故障が起きてもサービスを継続できる（問題を解くことができる），故障耐性のある分散プロトコルが重要である．

自己安定プロトコル（self-stabilizing protocol）とは，分散システムの任意の状況からプロトコルの実行を開始しても，やがて解状況に到達する（安定する）分散プロトコルである．この性質より，自己安定プロトコルは次の二つの特長をもつ．第1に，分散システムの初期化が不要である．第2に，任意の一時故障に耐性をもつ．一時故障とは，メモリの値の破壊や通信中のメッセージの改変などの一時的な故障である．自己安定プロトコルは，一時故障が生じたために分散システムがどのような状況に陥ってしまっても，計算機がプロトコルの実行を継続すれば，自動的に解状況に復帰して安定する．このように優れた性質をもつ自己安定プロトコルは，Dijkstra によって初めて提案され [4]，現在分散プロトコルの研究で最も盛んに研究されている分野の一つである．

自己安定プロトコルは一時故障に対する高度な故障耐性をもつ．しかし，自己安定プロトコルは，故障状況から再安定するまでの実行の間，再び故障が生じないことを仮定している．つまり，現実のシステムのように，一度障害を起こした計算機が，その後も断続的に障害を起こすような場合，自己安定プロトコルは解

† 奈良先端科学技術大学院大学情報科学研究科，生駒市
Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630–0101 Japan
†† 奈良先端科学技術大学院大学情報科学センター，生駒市
Information Technology Center, Nara Institute of Science and Technology, 8916–5 Takayama-cho, Ikoma-shi, 630–0101 Japan
††† 大阪大学大学院基礎工学研究科情報数理系専攻，豊中市
Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University, 1–3 Machikaneyama-cho, Toyonaka-shi, 560–8531 Japan
* 現在，高松大学経営学部
** 現在，名古屋工業大学電子情報工学科

状況に復帰して安定することを保証しない．このような断続的な障害を扱うには，永久故障の枠組みが必要となる．実システムにおける永久故障に対応するため，一時故障だけでなく，永久故障に対する故障耐性も有する自己安定プロトコルに関する研究も行われている [2], [10]〜[12]．本論文でも，永久故障に対する故障耐性を有する自己安定プロトコルについて考察する．

プロセスの永久故障としては，これまでに，ビザンチン故障，停止故障などの様々な故障モデルが考えられている．ビザンチン故障モデルは，故障プロセスの動作に仮定を設けない故障モデルであり，最も大きな故障クラスである．一方，停止故障モデルは，故障したプロセスはそれ以降，一切動作しなくなる故障モデルである．停止故障プロセスが 1 個でも存在すると，コンセンサス問題のような単純な問題でさえ解けないことが知られている [7]．これは，プロセスが停止故障していることと単に動作が遅いことが有界時間で区別できないことによる．また，Anagnostou ら [2] は，故障敏感（failure-sensitive）な問題のクラスを定義し，停止故障プロセスが 1 個でも存在すると，故障敏感な問題を解く自己安定プロトコルが存在しないことを示した．生成木構成問題は故障敏感な問題のクラスに属する．

生成木構成問題は基本的な分散問題の一つであり，経路情報の管理や同報通信など，様々な応用が存在する．通常の分散アルゴリズムの研究では，Gallager ら [9] などにより盛んに研究されてきた．また，自己安定プロトコルにおいては，Dolev ら [6] や Chen ら [3] をはじめ，様々な研究が行われている．なお，最適な自己安定生成木構成プロトコルは，Aggrawal ら [1] によって提案された，レジスタ通信モデル上で安定時間 $O(diam)$ のものである．$diam$ はネットワークの直径である．このプロトコルでは大域的なプロセス識別子を用いている．

これまで，永久故障を考慮した自己安定生成木構成プロトコルは提案されていない．本論文では，新たに非停止永久故障を導入することで，状態通信モデルにおいて，永久故障に耐性を有する自己安定生成木構成プロトコルを提案する．非停止永久故障は，故障プロセスが無限にしばしば，故障動作により状態変化する故障モデルである．停止を許さないという制限のもとで最も性質の悪い故障であるといえる．本論文では，非停止永久故障のもとで生成木構成問題を解く自己安定プロトコルを提案するが，これは，自己安定生成木構成問題の可解性にとって，停止故障が致命的な影響を与えることを意味する．

本論文で提案するプロトコルは，故障プロセス数に対して制限を置いていない．したがって，永久故障に対する優れた故障耐性を実現しているといえる．また，大域的なプロセス識別子は用いない．その代わりに，故障しない根プロセスを仮定する．これはネットワークが完全に対称であると，決定性プロトコルで問題を解くことが不可能なため導入した仮定であるが，故障耐性の観点からは欠点といえる．また停止故障と区別するため，故障プロセスの状態変化が，すべての正常な隣接プロセスによって無限にしばしば観測されるという仮定を設けている．提案するプロトコルの安定時間は $n(n/2+\mathcal{F})d$ である．n はプロセス数，d はネットワークの最大次数，$\mathcal{F}$ は故障プロセスの状態変化が観測されるまでの時間の上界（定数）である．

本論文の構成は以下のとおりである．**2.** でモデル及び生成木構成問題の定義を行う．**3.** では，非停止永久故障の存在下で生成木構成問題を解く自己安定プロトコルを提案し，その正当性を証明する．また，提案するプロトコルが安定するまでの時間計算量を評価する．最後に **4.** で結論と今後の課題について述べる．

2. 諸 定 義

2.1 分散システム

分散システムは n 個のプロセスとそれらを相互に結ぶ通信リンクからなり，無向グラフ $G=(V,E)$ によって表される．頂点集合 $V=\{p_0,p_1,\ldots,p_{n-1}\}$ はプロセスの集合を表し，辺集合 $\mathcal{E}$ は通信リンクの集合を表す．ここで $(p_i,p_j)\in E$ であるとき，プロセス p_i はプロセス p_j に隣接するという．プロセス p_i の隣接プロセスの集合を $N_i(\subseteqq V)$ と表す．p_i は，隣接プロセス $p_j\in N_i$ を局所的なポート番号を用いて識別する．すなわち，ポート番号の集合 $\widehat{N_i}=\{0,1,\ldots,|N_i|-1\}$ に対して，1 対 1 対応の関数 $\Lambda_i: N_i\rightarrow\widehat{N_i}$ によって，各隣接プロセスに対応するポート番号が定まっている．

本論文では，簡単のため，各プロセスは隣接プロセスの状態を直接読むことができる状態通信モデルを対象とする．ただし本論文で提案するプロトコルは，隣接プロセスが共有レジスタを用いて通信するレジスタ通信モデル上のプロトコルへと容易に変換することができる．

各プロセス p_i は状態機械であり，状態集合 S_i，状態遷移関数 α_i の組 (S_i,α_i) で定義される．状態通

信モデルでは，プロセス p_i の状態遷移関数 α_i は $\alpha_i : S_i \times (\prod_{p_j \in N_i} S_j) \to S_i$ である(注1)．

2.2 分散システムの実行

分散システム全体の大域的な状況は，全プロセスの状態の n 項組で表す．つまり，すべての可能な状況の集合を $\mathcal{C}$ とすると，$\mathcal{C} = \prod_{p_i \in V} S_i$ である．ある状況 $c \in \mathcal{C}$ においてプロセスの部分集合 $Q \subseteqq V$ が同時に動作し，システムが状況 c から状況 $c'(\in \mathcal{C})$ に変化したとする．これを，$c' = \sigma(c)$ と表す．ここで，$\Delta = (\alpha_0, \ldots, \alpha_{n-1})$ とすると，$\sigma = (\Delta, Q)$ と表され，これをステップと呼ぶ．上記において，$c = (s_0, \ldots, s_{n-1})$，$c' = (s'_0, \ldots, s'_{n-1})$ とすると，$p_i \in Q$ ならば $s'_i = \alpha_i(s_i; c/N_i)$ である．ここで c/N_i は，状況 c からプロセス $p_{j_\ell} \in N_i$ $(0 \leqq \ell \leqq |N_i| - 1)$ の状態 s_{j_ℓ} を集めた $|N_i|$ 項組 $(s_{j_0}, s_{j_1}, \ldots, s_{j_{|N_i|-1}})$ を表す．一方，$p_i \notin Q$ ならば $s'_i = s_i$ である．

スケジュールはプロセスの空ではない部分集合の無限系列 $Q^0, Q^1, \ldots$ $(Q^\ell \subseteqq V)$ である．状況 c^0 とスケジュール $Q^0, Q^1, \ldots$ が与えられたとき，c^0 から始まるスケジュール $Q^0, Q^1, \ldots$ によるシステムの実行 $\mathcal{E}$ は，状況の無限系列 $c^0, c^1, \ldots$ で表される．ただし各 ℓ について $c^\ell = (s^\ell_0, \ldots, s^\ell_{n-1})$ とすると，$c^{\ell+1} = \sigma^\ell(c^\ell)$, $\sigma^\ell = (\Delta, Q^\ell)$ を満たす．$p_i \in Q^\ell$ のとき，プロセス p_i はステップ σ^ℓ において動作したと呼び，$s^\ell_i \neq s^{\ell+1}_i$ のとき，p_i は状態変化したと呼ぶ．ここで実行 $\mathcal{E} = c^0, c^1, \ldots$ における状況 c^0 を実行 $\mathcal{E}$ の初期状況と呼ぶ．

本論文では自己安定プロトコルについて考察するので，初期状況に対し仮定を置かない．また，無限スケジュールを考えるが，すべてのスケジュールが公平であると仮定する．すなわち，各プロセスはスケジュールに無限にしばしば現れるものとする．

以下では，実行 $\mathcal{E} = c^0, c^1, \ldots$ の部分系列 $c^k, c^{k+1}, \ldots, c^{k'}$ を，実行断片と呼び $frag(\mathcal{E}; k, k')$ と表す．また，$frag(\mathcal{E}; k, \infty)$ は接尾部 $c^k, c^{k+1}, \ldots$ を表す．

2.3 非停止永久故障

本論文では，プロセスの故障を考える．故障プロセスとは，分散システムで定められた状態遷移関数に従わない状態遷移を行うプロセスである．形式的には，故障プロセスを以下のように定義する．

状況 c^0 とスケジュール $Q^0, Q^1, \ldots$ に対して，状況の無限系列 $\mathcal{E} = c^0, c^1, \ldots$ を考える．ここで，$c^\ell = (s^\ell_0, \ldots, s^\ell_{n-1})$ と表し，$p_i \notin Q^\ell$ ならば $s^{\ell+1}_i = s^\ell_i$ とする．$p_i \in Q^\ell$ に対して，$s^{\ell+1}_i \neq \alpha_i(s^\ell_i; c^\ell/N_i)$ のとき，p_i は実行 $\mathcal{E}$ のステップ σ^ℓ に故障動作したという．実行 $\mathcal{E}$ において故障動作したプロセスを故障プロセスという．以下では，故障プロセスの集合を F と表す．

これまでに，様々な故障モデルが考察されている．ビザンチン故障は，故障動作に関して何も仮定しない故障モデルであり，最も大きな故障クラスである．また，停止故障は，実行のある時点以降，状態が変化しない故障モデルである．停止故障プロセスが一つでも存在すると，コンセンサス問題のような単純な問題でさえ解けないことが知られている [7]．これは，プロセスが停止故障していることと単に動作が遅いことが有界時間で区別できないことによる．そこで本論文では，停止を許さない故障のモデルとして，故障プロセスが無限にしばしば，故障動作により状態変化する非停止永久故障を導入する．非停止永久故障は，以下のように定義される．

［定義 1］（非停止永久故障）

スケジュール $Q^0, Q^1, \ldots$ に対して，状況の無限系列 $\mathcal{E} = c^0, c^1, \ldots$ を考える．ここで，$c^\ell = (s^\ell_0, \ldots, s^\ell_{n-1})$ と表し，$p_i \notin Q^\ell$ ならば $s^{\ell+1}_i = s^\ell_i$ とする．p_i に対して，$p_i \in Q^\ell$ かつ $s^{\ell+1}_i \neq \alpha_i(s^\ell_i; c^\ell/N_i)$ かつ $s^{\ell+1}_i \neq s^\ell_i$ なる ℓ が無限個存在するとき，p_i は実行 $\mathcal{E}$ において非停止永久故障したという． □

本論文では，故障プロセスの故障として非停止永久故障のみを考える．ただし，故障プロセスが無限にしばしば，故障動作により状態変化したとしても，隣接プロセスがその状態変化を観測できなければ，この故障プロセスの故障は停止故障と同じになってしまう．例えば，正常なプロセス $p_i \in N_f$ が 2 度動作する間に，故障プロセス p_f が $s^\ell_f \to s^{\ell'}_f \to s^{\ell''}_f$ のように複数回状態遷移したとする．$s^\ell_f \neq s^{\ell'}_f$ であっても，$s^\ell_f = s^{\ell''}_f$ であれば，p_i は故障プロセスの状態変化を観測できない．そこで以下では，故障プロセスの故障動作による状態変化が任意の正常な隣接プロセスによって無限にしばしば観測されるという仮定を設ける．ここで，状態変化の観測は次のように定義される．

［定義 2］（状態変化の観測）

任意の実行 $\mathcal{E} = c^0, c^1, \ldots$ を考える．以下を満たす実行断片 $frag(\mathcal{E}; k, k')$ $(k < k')$ が存在するとき，ス

(注1)：厳密には，p_i は隣接プロセスをポート番号で識別するので，$\alpha_i : S_i \times \left(\prod_{j \in \widehat{N_i}} S_{j'} \right) \to S_i$（ただし，$p_{j'} = \Lambda_i^{-1}(j)$）である．

テップ $\sigma^{k'-1}$ において，プロセス $p_j \in N_i$ がプロセス p_i の状態変化を観測するという．

- $s_i^k \neq s_i^{k'-1}$
- $p_j \in Q^k$, $p_j \in Q^{k'-1}$, $p_j \notin Q^\ell$ $(k < \ell < k'-1)$ □

2.4 非停止永久故障下の自己安定生成木構成問題

分散システム $G=(V,E)$ において，非停止永久故障プロセスの集合を F とする．本論文では，G から故障プロセスを取り除いたネットワーク $G-F$ の生成木を構成する自己安定プロトコルを提案する．ここでは，$G-F$ の生成木を構成する自己安定プロトコルを定義する．ただし以下では，プロセス p_0 が根として指定されているものとする．また，p_0 は故障しないと仮定し，ネットワーク $G-F$ は連結であると仮定する．

各プロセス p_i は変数 $parent_i \in \widehat{N_i} \cup \{\perp\}$ をもつ．状況 c における変数 $parent_i$ の値を $parent_i(c)$ と表し，$T(c)=(V,A(c))$ を，V を頂点集合，$A(c)=\{(p_i,p_j)|parent_i(c)=\Lambda_i(p_j)\}$ を有向辺集合とするグラフとする $(\Lambda_i : N_i \to \widehat{N_i})$．また $T(c)-V'$ は，部分グラフ $(V-V', A(c)-(V'\times V'))$ を表す $(V' \subseteqq V)$．

［定義 3］ 頂点の部分集合 $V'(\subseteqq V-\{p_0\})$ が与えられたとき，$T(c)-V'$ が根プロセス p_0 を根とする木であるとは，以下の条件を満たすことをいう．

- 根プロセス p_0 の出次数は 0 $(parent_0(c)=\perp)$.
- 任意の頂点 $p_i(\in V-V'-\{p_0\})$ の出次数は 1 $(parent_i(c) \neq \perp)$.
- $T(c)-V'$ において，任意の頂点 $p_i(\in V-V')$ から根 p_0 に到達可能． □

［定義 4］（自己安定生成木構成プロトコル）
任意の故障プロセス集合 F に対する，プロトコル $\mathcal{A}$ の任意の実行 $\mathcal{E}$ が次の条件を満たす接尾部 $frag(\mathcal{E};k,\infty)$ をもつとき，プロトコル $\mathcal{A}$ を非停止永久故障耐性を有する自己安定生成木構成プロトコルという．

- 任意の状況 c^ℓ $(\ell \geqq k)$ において，$T(c^\ell)-F$ が根プロセス p_0 を根とする木．
- 任意の状況の組 $c^\ell, c^{\ell'}$ $(k \leqq \ell \leqq \ell')$ について，$T(c^\ell)-F=T(c^{\ell'})-F$. □

2.5 ラウンド

生成木を構成して安定するまでの時間計算量を安定時間という．非同期式分散システムにおいては，プロセスがいつ動作するかはわからない．そのままでは時間計算量が評価できないため，状態通信モデルやレジスタを用いた共有メモリ通信モデルでは，通常，1 単位時間（ラウンド）に各プロセスが少なくとも 1 回動作するという仮定を設ける．本論文でも以下に定義するラウンド数を用いて，安定時間を評価する．

［定義 5］（ラウンド）
任意の実行 $\mathcal{E}$ が与えられたとき，

- 第 0 ラウンドは，初期状況から始まる実行断片において，各プロセスが少なくとも 1 回はスケジュールに現れるような最小の実行断片 $frag(\mathcal{E};0,\ell_0)$ である．
- 第 k ラウンド $frag(\mathcal{E};\ell_{k-1},\ell_k)$ が定義されたとき，第 $k+1$ ラウンドは状況 c^{ℓ_k} から始まる実行断片において，各プロセスが少なくとも 1 回はスケジュールに現れるような最小の実行断片 $frag(\mathcal{E};\ell_k,\ell_{k+1})$ である． □

なお，メッセージ交換モデルでは，類似の評価尺度として理想時間計算量を用いるのが一般的である．理想時間計算量では，プロセスの動作時間は無視し，メッセージ伝送遅延がたかだか 1 単位時間であると仮定する．

3. プロトコル

本章では，非停止永久故障耐性を有する自己安定生成木構成プロトコルを示し，その正当性の証明と安定時間の評価を行う．

3.1 非停止永久故障耐性を有する自己安定生成木構成プロトコル

各正常プロセス p_i の状態遷移関数 α_i は以下で示される手続き全体により表される．つまり，正常プロセス p_i がスケジュールに現れるとき，直前の状況における隣接プロセスの状態（諸変数の値）に従って 1 ステップで手続き全体が処理され，新たな p_i の状態が決められる．

各プロセス p_i は，p_i における局所的なポート番号の集合 $\widehat{N_i}$ を定数としてもつ．また，**2.4** において定義された変数 $parent_i$（親へのポート番号を格納）に加え，$dist_i$ と old_i の 2 変数をもつ．変数 $dist_i$ には安定状況において根からの距離が格納され，変数 old_i は p_i が前回動作したときの $parent_{parent_i}$ の値を保持するための変数である．

各プロセスは自分が根であるか否かを関数 $Root$ を用いることで判別できる．根以外の各プロセス p_i は，$dist_{parent_i}$ 及び $parent_{parent_i}$ の値を前回動作したときの値と比較することで，親が状態変化したかどうかを観測する．親が状態変化したことを観測すると，p_i は現在の親とは異なる隣接プロセスを新たな親として

選ぶ．なぜなら，状態変化したプロセスは故障プロセスである可能性があるからである．なお，新しい親を選ぶ際には，あらかじめ決められた順序で繰返し隣接プロセスを選ぶ関数 $RRobin$ を使用する．これにより，p_i の隣接プロセスに故障プロセスが存在し，無限にしばしば状態変化が観測される場合には，故障プロセスを避けて親を選ぶことになる．

定数 $\widehat{N}_i$　隣接プロセスを表すポート番号の集合．
変数 $dist_i$　根からの距離．
　　old_i　前回の親の親の値を保持するための変数．
関数 $Root$　プロセスが根 p_0 であれば真，根以外であれば偽を返す．
　　$RRobin$　与えられた集合 X からラウンドロビンで集合の要素を返す．つまり X を順序集合とみなし，ℓ 回目に呼ばれたとき，$\ell \bmod |X|$ 番目の要素を返す．

プロトコル
if $(Root())$
　$parent_i := \perp$;
　$dist_i := 0$;
else if $((old_i, dist_i) \neq (parent_{parent_i}, dist_{parent_i}+1))$
　$parent_i := RRobin(\widehat{N}_i)$;
　$(old_i, dist_i) := (parent_{parent_i}, dist_{parent_i} + 1)$;

3.2 正　当　性

本節では提案したプロトコルが問題の解条件を満たすことを示す．問題の定義において導入した記法 $parent_i(c), T(c)$ に加え，$dist_i(c), old_i(c)$ は，それぞれ状況 c における各プロセス p_i の変数 $dist_i, old_i$ の値を表すものとする．

［補題 1］ 任意の実行 $\mathcal{E} = c^0, c^1, \ldots$ を考える．状況 c^k におけるグラフ $T(c^k)$ が有向閉路を含み，有向閉路に正常プロセスが含まれるならば，その有向閉路に含まれる少なくとも 1 個の正常プロセスが接尾部 $frag(\mathcal{E}; k, \infty)$ において状態変化し親を変更する．

（証明） $T(c^k)$ に含まれる有向閉路を，プロセスの系列を用いて $p_{j_0}, \ldots, p_{j_{m-1}}, p_{j_m}(= p_{j_0})$ と表す．ただし，$parent_{j_\ell}(c^k) = \Lambda_{j_\ell}(p_{j_{\ell+1}})$ $(0 \leqq \ell \leqq m-1)$ とする．有向閉路に故障プロセスが含まれる場合は，故障の定義とプロトコルより，故障プロセスを親とする正常プロセス p_{j_ℓ} はいずれ状態変化する．このとき，ネットワーク $G - F$ の連結性の仮定より p_{j_ℓ} の次数は 2 以上なので，p_{j_ℓ} は親を変更する．有向閉路に含まれるすべてのプロセスが正常プロセスの場合，各プロセス $p_{j_\ell}(\notin F)$ $(0 \leqq \ell \leqq m-1)$ が c^k 以降状態変化しないのであれば，各 ℓ について $dist_{j_\ell}(c^k) = dist_{j_{\ell+1}}(c^k) + 1$ が成立．つまり $dist_{j_0}(c^k) > dist_{j_{m-1}}(c^k) > dist_{j_0}(c^k)$ が成立することになり矛盾する．したがって，$dist_{j_\ell}(c^k) \neq dist_{j_{\ell+1}}(c^k) + 1$ であるようなプロセス p_{j_ℓ} が存在し，次に動作するとき状態変化する．ここで，p_{j_ℓ} の次数が 2 以上か，または，根である場合，p_{j_ℓ} が親を $p_{j_{\ell+1}}$ 以外に変更する．p_{j_ℓ} の次数が 1 で，かつ，根ではない場合．p_{j_ℓ} の次数が 1 となるのは長さ 2 の有向閉路 $p_{j_0}, p_{j_1}, p_{j_0}$ のときのみである．p_{j_ℓ} は $dist_{j_\ell}$ の値のみを変更し，$p_{j_{\ell-1}}$ が親を p_{j_ℓ} 以外に変更する． □

［補題 2］ 任意の実行 $\mathcal{E}$ において，無限にしばしば状態変化するプロセス集合を M と表す．実行 $\mathcal{E}$ に接尾部 $frag(\mathcal{E}; k, \infty)$ が存在し，任意の $\ell \geqq k$ について，$T(c^\ell) - M$ は連結である．

（証明） $V - M$ に含まれるプロセスが状態変化しない接尾部 $frag(\mathcal{E}; k, \infty)$ を考える．背理法により $T(c^\ell) - M(\ell \geqq k)$ が 2 個以上の連結成分を含むと仮定し，根プロセス p_0 を含まない連結成分の一つを $\overline{T_r} = (\overline{V_r}, \overline{A_r})$ と表す．補題 1 より $\overline{T_r}$ は閉路を含まない．一方，$\overline{T_r}$ の各プロセスの出次数が 1 だから，$T(c^\ell)$ において，$p_i \in \overline{V_r}$ かつ $p_j \in V - \overline{V_r}$ である p_i, p_j の組のうち，$(p_i, p_j) \in A(c^\ell)$ であるようなものが存在し，$\overline{T_r}$ が連結成分なので $p_j \in M$ である．p_i の次数が 2 以上の場合，プロトコルより，いずれ p_i は $parent_i$ を変更するので $p_i \notin M$ に矛盾する．p_i の次数が 1 の場合，ネットワーク $G - F$ の連結性の仮定より $p_j \notin F$ である．プロトコルより p_j はいずれ p_i を親とし，状態変化しなくなる．これは $p_j \in M$ に矛盾する．

したがって $T(c^\ell) - M$ は連結であることがいえる． □

［補題 3］ 任意の実行 $\mathcal{E}$ において，無限にしばしば状態変化するプロセス集合を M と表す．このとき $F = M$ が成立する．

（証明） 非停止永久故障の定義より $F \subseteq M$ である．$V - M$ に含まれるプロセスが状態変化しないような E の接尾部 $frag(\mathcal{E}; k, \infty)$ を考え，背理法により $(V - F) \cap M \neq \emptyset$ と仮定する．仮定より $G - F$ は連結なので，$p_i \in M - F$ かつ $p_j \in V - M$ であるようなプロセスの組 p_i, p_j $((p_i, p_j) \in E)$ が存在する．仮定より p_i は無限にしばしば親を変更するが，プロト

コルより，いずれ $V-M$ に属するプロセスを親とし状態変化しなくなることになり矛盾する．したがって，$(V-F)\cap M=\emptyset$ であることがいえ，$F=M$ がいえる． □

補題 2 及び補題 3 から，定理 1 がいえる．

［定理 1］ 任意の実行 $\mathcal{E}$ において，いずれグラフ $T(c)-F$ は根プロセス p_0 を根とする木となり安定する． □

3.3 安定時間

次に提案プロトコルが安定するまでの時間計算量である安定時間を評価する．故障プロセスの状態変化が観測されない間，正常なプロセスが故障プロセスを避けて木を構成することができないことは自明である．時間計算量を評価するため，各故障プロセス p_f について，p_f のすべての隣接プロセスが p_f の状態変化を観測するまでに要するラウンド数の上界を $\mathcal{F}$ と仮定する．定数 $\mathcal{F}$ は故障の見つかりにくさを示す指標ということができる．またネットワークの最大次数を d とする．

［補題 4］ 任意の実行 $\mathcal{E}$ において，2 個のプロセス $p_i, p_j \notin F$ を考える．p_j が状態変化した直後の状況を c^ℓ とし，c^ℓ は第 $k (k \geqq 2)$ ラウンドに属するとする．$T(c^\ell)$ において $(p_i,p_j)\in A(c^\ell)$ ならば，p_i は c^ℓ 以降第 $k+1$ ラウンド終了までに状態変化する．

（証明） 実行 $\mathcal{E}$ において，$\ell' < \ell < \ell'', p_i \in Q^{\ell'}, p_i \in Q^{\ell''-1}$ であるような最小の実行断片 $frag(\mathcal{E};\ell',\ell'') = c^{\ell'}, c^{\ell'+1}, \ldots, c^{\ell''}$ を考える．$k \geqq 2$ なので，$p_i \in Q^{\ell'}$ であるような ℓ' は存在する．また，$\sigma^{\ell''-1} = (\Delta, Q^{\ell''-1})$ は第 k または第 $k+1$ ラウンドに含まれることになる．

$c^{\ell'}$ において p_j の親が p_i でない場合と p_i である場合に分けられる．

- $parent_j(c^{\ell'}) = \Lambda_j(p_{j'})$ $(p_{j'} \neq p_i)$ の場合．

場合分けの仮定より $|N_j| \geqq 2$ なので，補題の仮定より，$parent_j(c^{\ell-1}) \neq parent_j(c^{\ell})$ である．$parent_j(c^{\ell'}) = parent_j(c^{\ell''-1}) = \Lambda_j(p_{j'})$ と仮定すると，プロトコルより，実行断片 $frag(\mathcal{E};\ell'+1,\ell''-1)$ において，$parent_j(c^{\ell'''}) = \Lambda_j(p_i)$ である状況 $c^{\ell'''}$ $(\ell' < \ell''' \leqq \ell''-1)$ が存在することになる．実行断片 $frag(\mathcal{E};\ell''',\ell''-1)$ において p_i は状態変化しないので，プロトコルより p_j も状態変化しない．つまり，$parent_j(c^{\ell''-1}) = \Lambda_j(p_i)$ となり矛盾する．したがって，$parent_j(c^{\ell'}) \neq parent_j(c^{\ell''-1})$ がいえ，p_i は $\sigma^{\ell''-1}$ において，p_j の状態変化を観測可能である．

- $parent_j(c^{\ell'}) = \Lambda_j(p_i)$ の場合．$parent_j(c^{\ell'}) = parent_j(c^{\ell''-1}) = \Lambda_j(p_i)$ と仮定する．p_i は $\sigma^{\ell'}$ で p_j を親としてから，$frag(\mathcal{E};\ell'+1,\ell''-1)$ において動作しないので，$dist_j(c^{\ell'})+1 = dist_i(c^{\ell'+1})$ が成立する．ここで，実行断片 $frag(\mathcal{E};\ell',\ell''-1)$ において，p_j がステップ $\sigma^{\ell-1}$ の 1 回のみ動作し，かつ，$\ell=\ell'+1$ の場合は，補題の仮定から $s_j^{\ell-1} \neq s_j^{\ell}$ なので，$dist_j(c^{\ell'}) \neq dist_j(c^{\ell''})$ が成立する．それ以外の場合は p_j は $frag(\mathcal{E};\ell'+1,\ell''-1)$ で動作し p_i を親とするので，$dist_j(c^{\ell''-1}) = dist_i(c^{\ell'+1})+1$ が成立し，したがって $dist_j(c^{\ell'})+2 = dist_j(c^{\ell''})$ がいえる．どちらの場合も $dist_j(c^{\ell'}) \neq dist_j(c^{\ell''})$ がいえる．したがって，p_i は $\sigma^{\ell''-1}$ において，p_j の状態変化を観測可能である．

いずれの場合も p_i は $k+1$ ラウンド終了までに p_j の状態変化を観測可能であり，p_i は状態変化する． □

［定理 2］ $V-F$ に属するプロセスが状態変化しなくなるまでに要するラウンド数は，たかだか $n(n/2+\mathcal{F})d$ である．(d はネットワークの最大次数)

（証明） 実行 $\mathcal{E}$ の $V-F$ が状態変化しない接尾部（定理 1 により定義可能）において，各状況 c におけるグラフを $T(c)-F=T$ とおく．また，T において根 p_0 からの距離が h であるプロセス集合を V_h と表し，V_h のプロセスが第 $h(n-h/2+\mathcal{F})d$ ラウンド以降は状態変化しないことを，h による帰納法により証明する．初期状況からたかだか 1 ラウンドで根 $r \in V_0$ は状態変化しなくなる．$\bigcup_{\ell=0}^{h-1} V_\ell$ に含まれる各プロセスが状態変化しない接尾部 $frag(\mathcal{E};k_h,\infty)$ において，プロセス $p_i \in V_h$ はたかだか d 回しか親を変更しない．ここで，p_i が ℓ $(1 \leqq \ell \leqq d)$ 回状態変化した直後の状況を $c^{k_h^\ell}$ とする．(便宜上 $k_h^0 = k_h$ とおく)

状況 $c^{k_h^\ell}$ から，p_i が 1 回状態変化するまでに要するラウンド数はたかだか $(n-h)+\mathcal{F}$ であるが，これは状況 $c^{k_h^\ell}$ において，p_i から親をたどって得られる経路を，以下の 3 通りの場合に分けることで証明される．

- 経路上に故障プロセス $p_f \in F$ を含む場合．故障プロセス p_f を親とする正常プロセスが p_f の状態変化を観測し，状態変化するまでに $\mathcal{F}$ ラウンド要する．補題 4 より，経路上の正常プロセス p_j が状態変化してから，それを親とする正常プロセス $p_{j'}$ が状態変化するまでたかだか 1 ラウンド要する．経路長はたかだか $n-h$ なので，p_i が状態変化するまで，たかだか $(n-h)+\mathcal{F}$ ラウンド要する．
- 経路上に F に属するプロセスを含まず，経路

が閉路に接続する場合．たかだか 1 ラウンドで閉路に含まれるプロセスのどれかが状態変化する．上の場合と同様の議論により，p_i が状態変化するまで，たかだか $n-h$ ラウンド要する．

- 経路上に F に属するプロセスを含まず，経路の終点が p_0 である場合．経路上のプロセスのうちたかだか 1 ラウンドで状態変化するプロセスが存在する場合，上と同様の議論により，p_i が状態変化するまで，たかだか $n-h$ ラウンド要する．経路上のすべてのプロセスが 1 ラウンド以内に状態変化しない場合，経路に含まれるプロセス $p_j \in V_{h-1}$ は状況 $c^{k_h^0}$ 以降状態変化しないので，p_j を親とするプロセスは状況 $c^{k_h^\ell}$ 以降状態変化しない．経路上のあるプロセス $p_{j'}$ が状態変化しないならば，$p_{j'}$ を親とするプロセス $p_{j''}$ は状況 $c^{k_h^\ell}$ 以降状態変化しない．したがって，経路上のどのプロセスも状況 $c^{k_h^\ell}$ 以降状態変化しないが，T の定義により経路は T に含まれることになる． □

［系 1］ 提案プロトコルは，非停止永久故障のもとで生成木を構成する自己安定プロトコルであり，たかだか $n(n/2+\mathcal{F})d$ ラウンドで安定する． □

4. むすび

本論文では新たな故障モデルとして非停止永久故障を定義した．非停止永久故障は，停止を許さないという制限のもとで最も性質の悪い故障であるといえる．そして，非停止永久故障のもとで生成木構成問題を解く自己安定プロトコルを提案した．これは，自己安定生成木構成問題の可解性にとって，停止故障が致命的な影響を与えることを意味する．

本論文では問題を解くにあたって，故障プロセスの状態変化が，すべての正常な隣接プロセスによって無限にしばしば観測されるという仮定を設けている．停止故障と区別するためには，故障プロセスの状態変化を無限にしばしば観測する正常な隣接プロセスが，少なくとも 1 個は存在するという仮定が必要である．本論文の仮定を緩和し，この最低限の仮定のもとで自己安定生成木構成問題が解けるかどうか，また，そのままでは解けない場合には，故障プロセス数などについてどのような制限を設ければ問題が解けるかを明確にすることは今後の課題である．

更に，非停止永久故障のもとで，生成木構成問題以外の静的問題（解状況が変化しない問題）が解けるかどうか，相互排除問題などの動的問題（解状況が変化する問題）が解けるかどうかの考察も今後の課題である．

本論文で提案したプロトコルの安定時間はたかだか $n(n/2+\mathcal{F})d$ ラウンドである．ここで，n はプロセス数，d はネットワークの最大次数，$\mathcal{F}$ は故障プロセスの状態変化が観測されるまでの時間の上界である．永久故障を考慮しない最適な自己安定生成木構成プロトコル [1] の安定時間 $O(diam)$ と比較すると，安定時間を改善できる可能性が考えられる．ここで $diam$ はネットワークの直径である．計算量の下界を求め，安定時間を改善することも今後の課題である．

謝辞 日ごろより有用な御討論を頂いている奈良先端科学技術大学院大学の井上美智子助教授に深く感謝致します．本研究は一部，日本学術振興会・科学研究費補助金・基盤研究 C（2）（課題番号 12680349）の研究助成による．

文　献

[1] S. Aggarwal and S. Kutten, “Time-optimal self-stabilizing spanning tree algorithms,” Proc. 13th Conference on the Foundations of Software Technology and Theoretical Computer Science (FSTTCS), pp.15–17, 1993.

[2] E. Anagnostou and V. Hadzilacos, “Tolerating transient and permanent failures,” 7th Int. Workshop on Distributed Algorithms (LNCS725), pp.174–188, 1993.

[3] NS. Chen, HP. Yu, and ST. Huang, “A self-stabilizing algorithm for constructing spanning trees,” Information Processing Letters, vol.39, no.3, pp.147–151, 1991.

[4] E.W. Dijkstra, “Self stabilizing systems in spite of distributed control,” Commun. ACM, vol.17, pp.643–644, 1974.

[5] S. Dolev, Self-stabilization, MIT Press, 2000. ISBN 0-262-04178-2.

[6] S. Dolev, A. Israeli, and S. Moran, “Uniform self-stabilizing leader election,” Proc. 5th Workshop on Distributed Algorithms, pp.167–180, 1991.

[7] M.J. Fischer, N.A. Lynch, and M.S. Paterson, “Impossibility of distributed consensus with one faulty process,” Proc. 2nd. ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, pp.1–7, 1983.

[8] E. Fromentin, M. Raynal, and F. Tronel, “On classes of problems in asynchronous distributed systems with process crashes,” Proc. 19th International Conference on Distributed Computing Systems (ICDCS’99), pp.470–477, 1999.

[9] R. Gallager, P. Humblet, and P. Spira, “A distributed algorithm for minimum-weight spanning trees,” ACM Trans. Programming Languages and Systems, vol.5, no.1, pp.66–77, 1983.

[10] A. Gopal and K. Perry, “Unifying self-stabilization and fault-tolerance,” Proc. 12th Ann. ACM Symp. on Principles of Distributed Computing (PODC’92), pp.195–206, 1993.
[11] T. Masuzawa, “A fault-tolerant and self-stabilizing protocol for the topology problem,” Proc. 2nd. Workshop on Self-Stabilizing Systems, pp.1.1–1.15, Las Vegas, NV, 1995.
[12] H. Matsui, M. Inoue, T. Masuzawa, and H. Fujiwara, “Fault-tolerant and self-stabilizing protocols using an unreliable failure detector,” IEICE Trans. Inf. & Syst., vol.E83-D, no.10, pp.1831–1840, Oct. 2000.

（平成 13 年 9 月 5 日受付，14 年 2 月 7 日再受付）

浮穴　学慈　（正員）

平 9 阪大・理・物理卒．平 14 奈良先端科学技術大学院大学博士後期課程了．同年高松大学経営学部講師．分散アルゴリズムの研究に従事．博士（工学）．

片山　喜章　（正員）

平 2 阪大・基礎工・情報卒．平 6 同大大学院博士後期課程中退．同年奈良先端科学技術大学院大学情報科学研究科助手．平 7 同大情報科学センター助手．平 14 名工大電気情報工学科講師．分散プロトコルなどに関する研究に従事．博士（工学）．情報処理学会会員．

増澤　利光　（正員）

昭 57 阪大・基礎工・情報卒．昭 62 同大大学院博士後期課程了．同年同大情報処理教育センター助手．同大基礎工助教授を経て，平 6 奈良先端科学技術大学院大学情報科学研究科助教授．平 12 阪大基礎工学研究科教授，現在に至る．平 5 コーネル大客員準教授（文部省在外研究員）．分散アルゴリズム，並列アルゴリズム，テスト容易化設計，テスト容易化高位合成に関する研究に従事．工博．ACM，IEEE，EATCS，情報処理学会各会員．

藤原　秀雄　（正員：フェロー）

昭 44 阪大・工・電子卒．昭 49 同大大学院博士課程了．同大・工・電子助手，明治大・工・電子通信助教授，情報科学教授を経て，現在奈良先端大・情報科学教授．昭 56 ウォータールー大客員助教授．昭 59 マッギル大客員準教授．論理設計論，フォールトトレランス，設計自動化，テスト容易化設計，テスト生成，並列処理，計算複雑度に関する研究に従事．著書「Logic Testing and Design for Testability」(MIT Press) など．大川出版賞，IEEE Computer Society Outstanding Contribution Award，IEEE Computer Society Meritorious Service Award など受賞．情報処理学会会員．IEEE Computer Society Golden Core Member，IEEE Fellow．