I-04

乱数生成器故障時における自己安定トークン巡回分散乱択アルゴリズムの解析

Analysis of Token Circulation Distributed Algorithm with Faulty Random Number Generator

川合 慎治 [1]　大下 福仁 [1]　角川 裕次 [1]　増澤 利光 [1]

Shinji Kawai　Fukuhito Ooshita　Hirotsugu Kakugawa　Toshimitsu Masuzawa

1　はじめに

分散システムとは，多数の計算機 (以降，**ノード**) と，それらを接続する通信リンクで構成されるシステムである．近年，分散システムの大規模化が進んでおり，システム内のノードが故障することは避けられない．よって，分散システムに故障耐性を付加することは重要な課題である．分散システムに高い故障耐性を持たせる技法のひとつに，**自己安定** [1] という概念がある．自己安定な分散システムは，任意の初期状況から実行を開始しても，やがて所望の性質を満たした状況へ遷移し，以降その状況を維持し続ける．これにより，自己安定なシステムは，どのような一時故障 (記憶内容の喪失など) が発生しても，やがてその故障から自律的に復帰することができる．

分散システムにおいて，動作の一部に乱数を利用した**分散乱択アルゴリズム**が利用されている ([2]，[3]，[4]，[5]，[6]，[7])．乱数の利用により，乱数を利用しない決定性アルゴリズムでは非可解な問題を解くことができる場合がある．また，決定性アルゴリズムの計算量を乱数の利用により削減することができる．例えば，ノードに一意な ID が存在しないリングネットワークでは，決定性アルゴリズムでリーダー選挙問題を解くことができない．しかし，文献 [2] では，乱数を利用して，リーダー選挙問題を解くアルゴリズムが示されている．

これまでに提案されている分散乱択アルゴリズムは，各ノードが局所的な乱数生成器を利用できることを仮定している．さらに，多くの場合，乱数生成器が一様分布に従った乱数値を出力することを仮定し，アルゴリズムの性能解析が行われている．しかし，いくつかのノードにおいて，乱数生成器が故障する，もしくは侵入者に操作される場合，乱数生成器が一様分布に従った乱数値を生成しなくなる可能性が生じる．本稿では，このような乱数生成器の故障を **RNG**(Random Number Generator) **故障**と定義し，RNG 故障が発生する分散システムについて考察する．

既存の分散乱択アルゴリズム ([2]，[3]，[4]，[5]，[6]，[7]) は，いずれも RNG 故障は考慮していない．また，文献 [4] でも RNG 故障への興味が掲載されている．そこで本研究では，RNG 故障が存在するモデル (以降，**RNG 故障有りモデル**) において，アルゴリズムが所望の性質を満たすことができるのか，また，所望の性質を満たすとき，RNG 故障による性能への影響がどの程度かを評価することを目的とする．

本稿では，Herman の乱択自己安定トークン巡回アルゴリズム [2] の RNG 故障有りモデルにおける評価を行う．**乱択自己安定アルゴリズム**とは，乱数を利用した自己安定アルゴリズムである．また，**トークン巡回アルゴリズム**とは，**トークン**と呼ばれる特権をネットワーク中に巡回させるアルゴリズムである．Herman のアルゴリズムは奇数長の匿名単方向リングネットワークを対象とし，自己安定性により，任意の初期状況から実行を開始しても，やがてトークン巡回を実現することができる．Herman のアルゴリズムでは，トークン所有ノードは乱数値によってそのトークンを転送するかどうかを決めている．RNG 故障を考えない場合，どのような初期状況から実行を開始してもトークン数が 1 つになるまでに要する時間 (**収束時間**) の期待値は高々 $\frac{n^2}{2}\lceil \log n \rceil$，トークンが 1 つの状況に到達後，1 つのトークンが全ノードを巡回するまでに要する時間 (**巡回時間**) の期待値は $2n$ である (n はノード数)．

本稿における成果を示す．本研究は，はじめて RNG 故障に対する考察を行った研究である．RNG 故障を必ずトークンを巡回する乱数を生成する故障と定義し，RNG 故障有りモデルにおける Herman のアルゴリズムの評価を行った．まず，収束時間について，RNG 故障ノード数が $n-1$ である場合，その期待値が高々 $\frac{n(n+1)^2}{4}\lceil \log n \rceil$ であることを解析により示した．さらに，収束時間の期待値は RNG 故障ノード数によって異なると考えられるが，

[1]大阪大学，Osaka University

RNG 故障ノード数が $n-1$ の場合に最大であることをシミュレーションによって示した．次に，RNG 故障ノード数が n_f である場合，巡回時間の期待値が，$2n-n_f$ であることを解析により示した．

本稿の構成は以下の通りである．2 章では，本稿で用いる用語の定義を行う．3 章では，Herman のトークン巡回アルゴリズムを紹介する．4 章では，RNG 故障有りモデルにおいて，Herman のトークン巡回アルゴリズムの評価を行う．評価するのは，収束時間と巡回時間である．5 章では，本稿の結果をまとめる．

2 諸定義

2.1 分散システム

分散システム D は，2 項組 $D=(R,A)$ から成る．R はネットワークで，A はある目的を達成するためのアルゴリズムである．本稿では，以下に定義するように，トポロジーが単方向リングであるネットワーク R のもとで議論をする．

単方向リングネットワーク R は，2 項組 $R=(V,L)$ で定義される．V は計算機 (以降，**ノード**) の集合であり，L は単方向通信リンク (以降，**リンク**) の集合である．本稿では，ノード数を $n(=|V|)$ とし，ネットワーク R におけるノード u からノード v への距離 $d(u,v)$ を，u から v への (有効) 最短経路を構成するリンクの数によって定義する．また，本稿では，リングネットワークを次のように定義する．

- $V=\{v_0,v_1,\ldots,v_{n-1}\}$
- $L=\{(v_i,v_{(i+1) \bmod n})|0\leq i\leq n-1\}$

つまり，ノード v_0，v_1，…，v_{n-1} はこの順で単方向に並んでいる．以下簡単のため，特に言及する必要がないときは，ノードのインデックスに対する演算は n の法のもとで計算されるとし，$v_{(i+1) \bmod n}$ は単に v_{i+1} と表す．

ノード v_i と v_{i+1} とは，リンクで結ばれているので，互いに**隣接ノード**であるという．また，v_i にとって v_{i+1} を**前方ノード**といい，v_i から v_{i+1} への方向を**前方向**という．同様に，v_i にとって v_{i-1} を**後方ノード**といい，v_i から v_{i-1} への方向を**後方向**という．

アルゴリズム A は各ノードの動作を定義するものである．つまり，各ノードは状態機械でモデル化し，アルゴリズム A によって，各ノード v_i に**状態集合** S_i と**状態遷移関数** δ_i が与えられる．本稿では，δ_i に乱数を用いて出力を決定する関数を用いる．つまり，各ノード v_i は乱数生成器を所有しており，乱数生成器が生成する乱数 (乱数集合 RN_i) を用いて関数 δ_i の出力を決定する．各ノードは局所変数を持ち，この局所変数の値が各ノードの状態を表す．ここで，各ノードは隣接ノードの状態を参照することができるという通信モデルを仮定する．この通信モデルを**状態通信モデル**という．各ノード v_i は，隣接ノードのうち後方ノードの状態を読みとり，自分の状態，後方ノードの状態，乱数から次の状態を決定する．つまり，v_i の状態遷移関数は次のような関数である．

$$\delta_i\colon S_i\times S_{i-1}\times RN_i\to S_i$$

状態通信モデルは，エラーのない通信でしかもメッセージ遅延がない，あるいはそれと等価な機能がプロトコルの下位層で実現されていることを仮定している (状態通信モデルでは，ノードが隣接ノードの状態を知りたい時，その値をただちに読み出せる)．

また，本稿ではネットワーク中の全てのノードが同一の状態機械とする**匿名ネットワーク**を扱う．つまり，各ノードに一意な ID が存在せず，全て同じ状態空間で同じ遷移関数に従って動作を行う．

2.2 ネットワーク状況と実行

単方向リングネットワークのシステム状況は各ノードの状態の n 項組で表される．つまり，$\Gamma=S_0\times S_1\times\ldots\times S_{n-1}$ とすると，Γ が単方向リングネットワークシステムの全状況集合を表し，$c\in\Gamma$ がネットワーク状況 (以降，**状況**) を表す．

単方向リングネットワークのシステムの**実行** E は状況の系列 $c_0\ c_1\ \ldots$ として定義される．このとき，c_0 を**初期状況**と呼ぶ．本稿では，全ノードが同時に動作を行うモデルを考える．このモデルを**同期モデル**といい，このとき，各ノードは**同期**しているという．ある状況 $c=(s_0,s_1,\ldots,s_{n-1})$ に対する次の状況を $c'=(s'_0,s'_1,\ldots,s'_{n-1})$ とするとき，ノード v_i が生成する乱数を r_i としたならば，次が成立する．

$$s'_i=\delta_i(s_i,s_{i-1},r_i)\ \ (v_i\in V)$$

また，実行に要する時間を評価するために**ラウンド**を定義する．すべてのノードが 1 度だけ動作するのが 1 ラウンドである．本稿では，同期モデルにおける実行 $E=c_0c_1\ldots$ について，部分実行 c_0 を第 0 ラウンドとする．以降，部分実行 c_i を第 i ラウンドと定義する．また，状況 c から 1 ラウンド後の状況を $f(c)$，k ラウンド後の状況を $f^k(c)$ と表し，f を**状況遷移関数**と定義する．ここで，状況遷移関数 f に関して，ある状況 c からの遷移状況を $c_1=f(c)$，$c_2=f(c)$ としたとき，$c_1=c_2$ に

なるとは限らない．これは，f が決定的関数でないためである．

2.3 乱択自己安定トークン巡回アルゴリズム

はじめに，**乱択自己安定アルゴリズム**について説明する．アルゴリズム A が**正しい状況**の集合 C について乱択自己安定アルゴリズムであるとは，以下の条件が成り立つときである．

収束性： 任意の初期状況から始まる実行は，確率 1 で $c_i \in C$ となる状況 c_i へ到達する．

閉包性： 任意の正しい状況 $c_0 \in C$ を初期状況とする実行 $E = c_0c_1\ldots$ について，$c_i \in C$ が全ての i について成立する．

次に，**乱択自己安定トークン巡回アルゴリズム**について説明する．**トークン巡回アルゴリズム**とは，**トークン**と呼ばれる特権をネットワーク中に巡回させるアルゴリズムである．乱択自己安定トークン巡回アルゴリズムにおける正しい状況の集合 C は，以下のように定義される．

- $c_0 \in C$ ならば，c_0 を初期状況とする実行 $E = c_0c_1\ldots$ について，以下の条件が成立する．
  - ネットワーク中にただ 1 つのトークンが存在し，そのトークンを各ノードが無限回受け取る．

本稿では，評価指標として，**収束時間** T_{conv} と**巡回時間** T_{cir} を用い，それぞれ次のように定義する．

収束時間： 任意の初期状況から正しい状況へ到達するまでに要するラウンド数．

巡回時間： 任意の正しい状況から始まる実行において，トークンが全ノードを巡回するために要するラウンド数．

2.4 RNG 故障

RNG 故障とは，乱数生成器が故障する故障モデルである．まず，RNG 故障が生じたノードを**RNG 故障ノード**，RNG 故障が生じていないノードを**正常ノード**と定義する．

通常，正常ノード v_i の乱数生成器はある確率分布に従って乱数値 r_i を出力する．そのため，乱数値 r_i を用いる状態遷移関数 δ_i もある確率分布に従って次状態を出力する．一方，RNG 故障ノード v_j の乱数生成器はその確率分布が異なる．そのため，RNG 故障ノード v_j では，状態遷移関数 δ_j の出力が，アルゴリズムで想定されている確率分布と異なる確率分布で選択される．例えば，乱数生成器が 1 ビット生成するときを考える．正常ノード v_i は，確率 p で 0，確率 $1-p$ で 1 で乱数を生成するとしたとき，RNG 故障ノード v_j は，0 しか出力しない等が一例として挙げることができる．

本稿では，この故障を，RNG 故障と定義し，RNG 故障が存在するモデルを RNG 有り故障モデルと定義する．

3 Herman のトークン巡回アルゴリズム

Herman のトークン巡回アルゴリズム [2] (以降, HTCA) は乱数を利用した乱択自己安定アルゴリズムである．

はじめに，HTCA におけるシステムモデルについて説明する．HTCA は，ノードが同期しており，ノード数 n が奇数 ($n = 2m+1$，$m \geq 0$) の匿名単方向リングネットワーク上で動作する．また，各ノード v_i は乱数生成器を所有しており，各ラウンドで 1 ビットの乱数 r_i を利用する．

次に，アルゴリズムに関して説明する．各ノード v_i の状態 s_i は 1 ビット ($S_i = \{0,1\}$) で定義し，状況 $c = (s_0, s_1, \ldots, s_{n-1})$ は n ビットで定義する．状態遷移関数 δ_i は，次の通りである．

$$s'_i = \delta_i(s_i, s_{i-1}, r_i) = \begin{cases} s_{i-1} & \text{if} \quad s_i \neq s_{i-1} \\ r_i & \text{if} \quad s_i = s_{i-1} \end{cases}$$

このとき，正常ノード v_i の乱数生成器が生成する乱数値 r_i は，確率 $\frac{1}{2}$ で 0 が，確率 $\frac{1}{2}$ で 1 が出力される．ここで，$s_i = s_{i-1}$ であるノード v_i を**トークン所有ノード**と定義する．n は奇数のため，どのような状況でも奇数個 (従って，少なくとも 1 つ) のノードはトークンを所有する．

最後に，このアルゴリズムの動作を図を用いて説明する．ここで扱うネットワークは匿名だが，説明上区別のために図中ではノードに ID を示している．

図 1 は，ノード v_i がトークンを所有している図である．図 2 は，ラウンド k におけるトークンの移動が関数 $\delta_i(s_i, s_{i-1}, r_i)$ により決定することを表す図である．ここで，$s_i = s_{i-1}$ 時の関数 $\delta_i(s_i, s_{i-1}, r_i)$ の出力を r_i とし，ノードの状態がわからないものは x とする．$r_i = s_{i-1}$ のとき，トークンが移動し，$r_i \neq s_{i-1}$ のときトークンは滞在する．つまり，トークンは確率 $\frac{1}{2}$ で移動し，確率 $\frac{1}{2}$ で滞在する．図 3 は**トークンの衝突**を表す図である．トークンの衝突とは，ノード v_i がトークンを滞在させ，

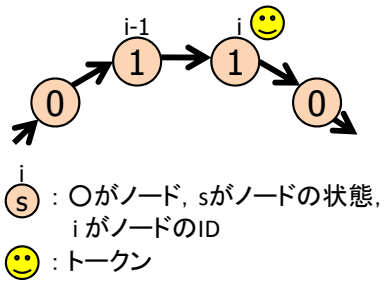

図 1: トークン所有例

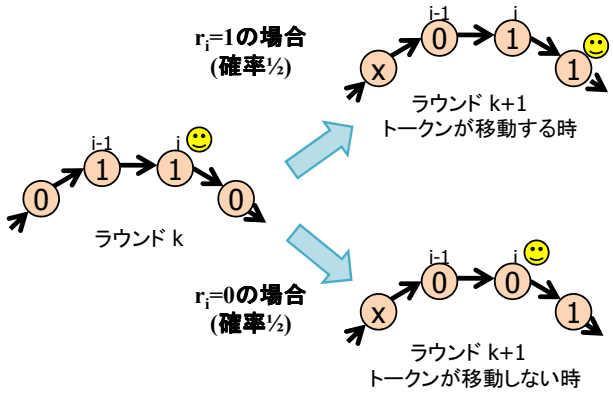

図 2: トークンの移動例

その後方ノード v_{i-1} がトークンを移動させることをいう．このとき，2 つのトークンは消滅する．トークンの衝突が起こらなければトークン数は変化せず，トークン数が増加することはない．

図 3 は，ノード v_i でトークンの衝突が起こる図である．図 4 はこのアルゴリズムの実行例を表す図である．トークンが複数個から 1 つになるまでを表している．ここでは，ラウンド k で，トークンは 3 個存在するが，ノード v_0 でトークンの衝突が起こり，ラウンド $k+1$ にはトークンが 1 つになる．

乱択自己安定トークン巡回アルゴリズムは確率 1 で正しい状況へ遷移しなければならない．次に，HTCA のアルゴリズムの正しさを検証する．

あるノード v_i がトークンを所有しているとき，$\frac{1}{2}$ でトークンを巡回させ，$\frac{1}{2}$ でトークンを滞在させるため，いずれ

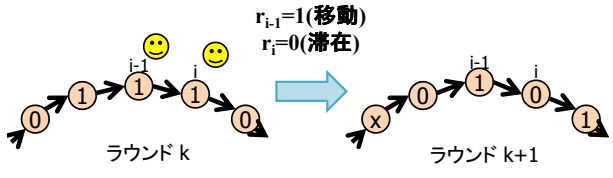

図 3: トークンの衝突例

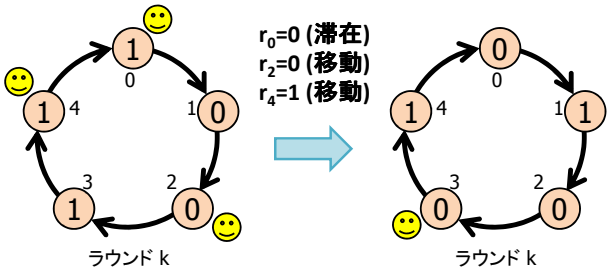

図 4: 実行例

トークンは巡回する．よって，各ノード $v_i(0 \leq v_i \leq n-1)$ において，いずれトークンが巡回するため，トークンが 1 つのとき，それが巡回し続ける．

状況 c におけるトークン数を $T(c)$ とする．$1 < T(c)$ である状況 c を考える．ノードは確率的にトークンを巡回させるため，いずれある確率でトークンの衝突が起こり，トークン数は 2 つ少なくなる．従って，どのような初期状況から実行を開始しても，いずれトークン数は 2 つずつ減少する．トークン数が奇数のため，トークン数はやがて 1 つになる．

4 評価

本節では，RNG 故障有りモデルにおける HTCA の収束時間と巡回時間の評価を行う．まず，本稿における RNG 故障の定義を行なう．本稿では，生成する乱数値の確率分布が大きく偏った場合について考察しようと考えた．そのため，RNG 故障ノードの乱数生成器は，HTCA において，必ずトークンを巡回させる乱数値を生成すると定義する．つまり，HTCA の状態遷移関数 $\delta_i(s_i, s_{i-1}, r_i)$ の計算において乱数値 r_i が必ず s_{i-1} となる．通常，アルゴリズムに乱数を用いる場合，期待値として評価を行う．また，収束時間は初期状況におけるトークン数によって変化する．そのため，収束時間を評価するとき，最悪時の初期状況 (すなわち，収束時間の期待値が最も長くなる状況) に対して，収束時間の期待値を評価する．以降では，RNG 故障ノード数が n_f のときの収束時間の期待値を $T_{conv}(n_f)$，巡回時間の期待値を $T_{cir}(n_f)$ と表記する．

4.1 節では，まず，RNG 故障ノード数が $n-1$ のときの収束時間の期待値 $T_{conv}(n-1)$ を求める．その後，シミュレーションにより，RNG 故障ノード数が $n-1$ のとき，収束時間の期待値 $T_{conv}(n-1)$ が RNG 故障有りモデルにおいて最大であることを示す．4.2 節では，RNG 故障ノード数が n_f のときの巡回時間の期待値 $T_{cir}(n_f)$ を導出する．

本稿では，HTCA の動作を変えることなく，HTCA の動作の本質的な動作だけを抽出した **Herman の一般モデル**を考える．Herman の一般モデルは以下の通りである．

正常ノードの動作： 確率 $\frac{1}{2}$ でトークンを移動させ，確率 $\frac{1}{2}$ でトークンを滞在させる．

RNG 故障ノードの動作： 確率 1 でトークンを移動させる．

トークンの衝突は，あるノード v_i がトークンを滞在させ，後方ノード v_{i-1} がトークンを移動させたときに起こり，このとき，衝突した 2 つのトークンは消滅する．Herman の一般モデルでは，ノード数 n は奇数とするが，状況 c におけるトークン数 $T(c)$ の偶奇は問わないとする．これは，Herman のモデルではあり得ないトークン数が 2 個の状況において解析を行い，その結果を Herman のモデルにおける解析に用いるためである．

4.1 収束時間の評価

4.1.1 $n_f = n-1$ の場合の解析

本節では，RNG 故障ノード数が $n-1$ のときの HTCA の収束時間の期待値 $T_{conv}(n-1)$ を導出する．まず，Herman の一般モデルにおいて，リングにトークンが 2 つある場合を考え，2 つのトークンが衝突するまでに要する時間 (以降，**衝突時間**) の期待値を求める．衝突時間を評価するとき，収束時間を評価するときと同様に，最悪時の初期状況 (すなわち，衝突時間の期待値が最も長くなる状況) に対して，衝突時間を評価する．以降では，RNG 故障ノード数が n_f のときの衝突時間の期待値を $T_{coll}(n_f)$ と表す．次に，$T_{conv}(n-1)$ が高々$T_{coll}(n-1)\lceil \log n \rceil$ であることを示す．

解析における諸定義を行う．トークンが 2 つ存在する状況 c を考える．v_0 を正常ノードとする．v_0 がトークンを所有しているとき，v_0 を**先頭ノード**，v_0 がトークンを所有していないとき，v_0 の後方向のうち，はじめてトークンを所有しているノードを先頭ノードとする．また，もう一方のトークンを所有するノードを**後尾ノード**とする．状況 c における 2 つのトークン間距離 $D(c)$ を，状況 c における先頭ノード v_h，後尾ノード v_t を用いて $d(v_t, v_h)$ と定義する ($D(c) = (h-t) \bmod n$，$0 \le D(c) < n$)．ただし，2 つのトークンが消滅した状況 c は，$D(c) = 0$ とする．図 5 はその一例である．

リングにトークンが 2 つある場合について，衝突時間の期待値 $T_{coll}(n-1)$ を導出する．以降，トークンを移動させるか，滞在させるかの処理を**トークンの処理**と呼ぶ．

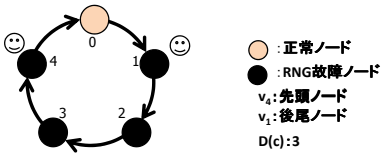

図 5: トークン間距離の例

トークンの処理によるトークン間距離の変化について考える．トークン所有ノードは先頭ノード v_h，後尾ノード v_t である．先頭ノード，後尾ノードについて考えられる状況は，先頭ノードが正常ノードであるか否かである (定義より，後尾ノードが正常ノードであることはない)．状況 c における先頭ノードを v_{h_c}，後尾ノードを v_{t_c} とし，状況 c の 1 ラウンド後の状況を c' とする．

- 状況 c で先頭ノード v_{h_c} が RNG 故障ノードの場合 ($h_c \neq 0$，$D(c) = h_c - t_c$)
  - 状況 c' で先頭ノードは v_{h_c+1}，後尾ノードは v_{t_c+1} となり，トークン間距離は $D(c') = (h_c+1) - (t_c+1) = h_c - t_c$ となる．つまり，このラウンドでトークン間距離は変化しない．
- 状況 c で先頭ノード v_{h_c} が正常ノードの場合 ($h_c = 0$，$D(c) = n - t_c$)
  - 先頭ノード v_0 がトークンを滞在させるとき，状況 c' で先頭ノードが v_0，後尾ノードが v_{t_c+1} となり，トークン間距離は $D(c') = n - t_c - 1$ となり，1 減る．
  - 先頭ノード v_0 がトークンを移動させるとき，状況 c' で先頭ノードが v_{t_c+1}，後尾ノードが v_1 となり，トークン間距離は $D(c') = (t_c+1) - 1 = t_c$ となる．

トークン間距離の変化が起こり得るのは，正常ノードがトークンの処理をするときである．そのため，はじめに，トークンの衝突に要する正常ノードのトークンの処理回数の期待値を導出する．このとき，正常ノードのトークンの処理に要する時間複雑度の単位として，**フェーズ**を用いる．ここで，実行 $E = c_0 c_1 \ldots$ について，状況 c_0 から正常ノードがトークンの処理を行う状況 c_i までを第 1 フェーズとし，以降，正常ノードがトークンの処理を行うまでを第 2 フェーズ，第 3 フェーズ，... とする．また，状況 c から 1 フェーズ後の状況を $F(c)$ と表し，状況 c から k フェーズ後の状況を $F^k(c)$ と表す．正常ノードがトークンを所有していなくても，少なくとも n ラウ

ンドの間に必ず 1 度はトークンの処理を行う．よって，1 フェーズは高々n ラウンドである．

まず，トークン間距離 D の変化が生じる確率を要素とする行列 A を定義する．行列 A は $n \times n$ の行列であり，行を 0 行から $n-1$ 行，列を 0 列から $n-1$ 列とする．行列 A の各要素 a_{ij} は $D(c)=i$ の状況 c から，1 フェーズ後に，$D(F(c))=j$ の状況へ遷移する確率とする．このとき，0 列目はトークンが無い状況を表し，行列 A の 0 行目はトークンが無い状況になり実行が終了したことを表す．行列 A は以下の通りとなる．

$$A = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \frac{1}{2} & 0 & \cdots & \cdots & \cdots & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \cdots & 0 & \frac{1}{2} & 0 \\ \vdots & 0 & \ddots & & \iddots & 0 & \vdots \\ \vdots & \vdots & & & & \vdots & \vdots \\ \vdots & 0 & \iddots & & \ddots & 0 & \vdots \\ 0 & \frac{1}{2} & 0 & \cdots & \cdots & \frac{1}{2} & 0 \end{bmatrix}$$

以降，行列 A を用いてトークンの衝突に要するフェーズ数の導出を行う．ここで，I を単位行列とする．文献 [2] より，補題 1 が成立する．

補題 1 $\sum_{j=0}^{\infty} jA^j = (I-A)^{-1}A(I-A)^{-1}$ □

次に補題 2 を導出する．

補題 2 リングに 2 つトークンが存在するとき，トークンの衝突に要するラウンド数の期待値は $\frac{n(n+1)^2}{4}$ である．

証明　e_i を $1 \times n$ のベクトルとし，単位行列 I の i 行目と等しいベクトルとする．状況 c において，$i = D(c)$ とする．このとき，ベクトル e_iA の第 j 要素は，トークン間距離が $D(c)=i$ の状況からトークン間距離が $D(F(c))=j$ の状況へ遷移する確率を表す．同様に，e_iA^k の第 j 要素は，トークン間距離 $D(c)=i$ の状況から，k フェーズ後に，トークン間距離が $D(F^k(c))=j$ の状況へ遷移する確率を表す．トークン間距離が $D(c)=i$ の状況から，k フェーズ後，$D(F^k(c))=0$ の状況に遷移する確率 $P(D(F^k(c))=0)$ は $e_iA^ke_0^T$ となる．よって，$D(c)=i$ の状況から，$D(F^k(c))=0$ の状況に要するフェーズの期待値 $E(D(F^k(c))=0)$ は以下である．

$$E(D(F^k(c))=0) = \sum_{j=0}^{\infty} je_iA^je_0^T = e_i(\sum_{j=0}^{\infty} jA^j)e_0^T$$

補題 1 より，次の式を得る．

$$\begin{aligned} E(D(F^k(c))=0) &= e_i(\sum_{j=0}^{\infty} jA^j)e_0^T \\ &= e_i(I-A)^{-1}A(I-A)^{-1}e_0^T \end{aligned}$$

行列 $(I-A)^{-1}$ を求めると以下の通りとなる．

$$B = (I-A)^{-1} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & 2 & 1 & \cdots & \cdots & \cdots & \cdots & 1 \\ \vdots & \vdots & 3 & 2 & \cdots & \cdots & 2 & \vdots \\ \vdots & \vdots & \vdots & & & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & & & 2 & \vdots \\ \vdots & \vdots & 3 & \cdots & \cdots & \cdots & 3 & 1 \\ 1 & 2 & \cdots & \cdots & \cdots & \cdots & \cdots & 2 \end{bmatrix}$$

よって，

$$E(D(F^k(c))=0) = e_i(BABe_0^T)$$

となる．$h = BABe_0^T$ としたとき，h は $n \times 1$ のベクトルであり，i 行目の成分 h_i は以下の通りである．

$$\begin{aligned} h_i &= ni - i^2 + i \\ &= -(i - \frac{n+1}{2})^2 + \frac{(n+1)^2}{4} \end{aligned}$$

ここで，$E(D(F^k(c))=0) = e_ih = h_i$ である．衝突時間の期待値の最大は以下の式で表すことができる．

$$\max_i\{h_i\}$$

n は奇数なので，h_i の式より，$i = \frac{n+1}{2}$ のとき衝突に要するフェーズの期待値が最大となり，$\frac{(n+1)^2}{4}$ となる．1 フェーズは高々n ラウンドであるので，衝突時間の期待値は $T_{coll}(n-1) = \frac{n(n+1)^2}{4}$ ラウンドとなる．□

補題 3 収束時間の期待値 $T_{conv}(n-1)$ は高々$T_{coll}(n-1)\lceil \log n \rceil$ である．

証明　初期状況 $c(T(c) \geq 2)$ で，以下に述べる方法でトークンを 2 つ選択する．選択した 2 つのトークンを**色付きトークン**と呼び，それら以外のトークンを**色無しトークン**と呼ぶ．初期状況で，正常ノードの後方向のうち，一番近い (正常ノードも含む) トークンを色付きトークンとし，トークン所有ノードを v_{ch} とする (トークン所有ノードは色付きトークンの移動とともに変わる)．また，v_{ch} からもう一方の色付きトークン所有ノード (v_{ct}) までの前方向の経路上に存在するトークン数が $1 + \left\lfloor \frac{T(c)}{2} \right\rfloor$ (つまり，後方向の経路上に存在するトークン数が $1 + \left\lceil \frac{T(c)}{2} \right\rceil$) となるようにもう一方の色付きトークンを選択する．v_{ch} と v_{ct} の距離 $l(v_{ch}, v_{ct})$ を $d(v_{ct}, v_{ch})$

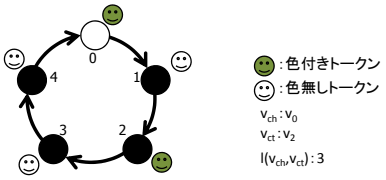

図 6: 色付きトークン間距離の例

とする ($l(v_{ch}, v_{ct}) = ch - ct \bmod n$)．図 6 は，色付きトークン間距離の例を表している．

以降，初期状況から色付きトークンが消滅する状況 (以降，**終了状況**) までに要するラウンド数が高々$T_{coll}(n-1)$であることを示す．そのため，ある状況 c から 1 フェーズ後，色付きトークンが衝突したとき，新たな色付きトークンの選択方法を示す．

状況 c から 1 フェーズ後の状況 c' において，色付きトークンが衝突したとき，新たな色付きトークン所有ノード $v_{ch'}$，$v_{ct'}$ を以下のように定義する．状況 c' において，色付きトークンが衝突したときを考える (トークンの衝突は正常ノードでのみ起こる)．

- 色付きトークン同士が衝突したとき，2 つのトークンは消滅し，終了状況とする．
- 色付きトークンと色無しトークンが衝突したとき，新たに色付きトークンを選択する．色付きトークンになる候補は，正常ノードの後方向のうち一番近いトークン (所有するノードを v_a)，もしくは，正常ノードの前方向のうち一番近いトークン (所有するノードを v_b) である．
  - 候補となるトークンのどちらかが色付きトークンのとき，終了状況とする．
  - 候補となるトークンのどちらも色無しトークンのとき，現存する色付きトークンとの距離が小さくなる方 (等しいとき，正常ノードの後方向のトークン) を色付きトークンとして選択する (図 7 参照)．

候補となるトークンのうち，v_a，v_b のどちらが色付きトークンを所有しても，候補となるトークンの選択方法より，色付きトークン間距離は減少する (つまり，$l(v_{ch'}, v_{ct'}) < l(v_{ch}, v_{ct})$)．よって，初期状況から終了状況までに要するラウンド数の期待値は高々$T_{coll}(n-1)$ である．

初期状況 c において，少なくとも，$1 + \lfloor \frac{T(c)}{2} \rfloor$ 個のトークンは色付きトークン間に存在し，終了状況では，色付きトークンは高々1 つ残る．つまり，終了状況では，色付きトークンが少なくとも 1 つ，色無しトークンが少なくとも $\lfloor \frac{T(c)}{2} \rfloor - 1$ 個消滅する．よって，終了状況のとき，トークン数 $T(c')$ は以下になる．

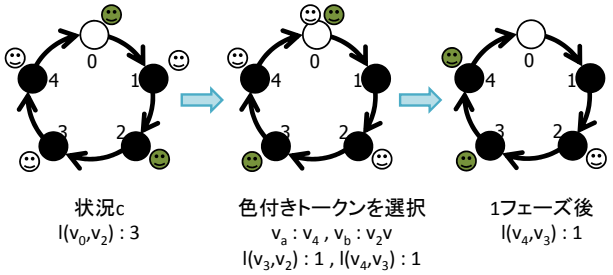

図 7: 色付きトークン衝突後の例

$$\begin{aligned} T(c') &\leq T(c) - (\left\lfloor \frac{T(c)}{2} \right\rfloor - 1) - 1 \\ &= \left\lceil \frac{T(c)}{2} \right\rceil \end{aligned}$$

初期状況から終了状況の間に，トークン数は半分以下になることがわかる．また，高々$\lceil \log n \rceil$ 回トークン数を半分にすればトークン数は 1 つになる．よって，収束時間の期待値 $T_{conv}(n-1)$ は高々$T_{coll}(n-1)\lceil \log n \rceil$ である．□

補題 3 より，Herman の一般モデルにおいて，トークンが複数存在するとき高々$\frac{n(n+1)^2}{4}\lceil \log n \rceil$ ラウンドで収束することを導出した．Herman の一般モデルではトークン数の偶奇を問わないが，Herman のモデルではトークン数を奇数としており，Herman の一般モデルにおける収束時間の期待値が Herman のモデルにおける収束時間の期待値ともなる．よって，定理 1 が導出される．

定理 1 *RNG 故障有りモデルにおける HTCA の収束時間の期待値 $T_{conv}(n-1)$ は高々$\frac{n(n+1)^2}{4}\lceil \log n \rceil$ である．*

4.1.2 シミュレーション

本節では，収束時間の期待値 $T_{conv}(n_f)$ が，$n_f = n-1$ のとき最大であることをシミュレーション結果より示す．収束時間に影響を及ぼすパラメータは，ノード数 n，RNG 故障ノード数 n_f，初期状況におけるトークン数である．ノード数 n を 99，初期状況におけるトークン数を 99 とし，RNG 故障ノード数 n_f を 0 から $n-1$ まで変化させ，シミュレーションを行う．このとき，RNG 故障ノード数 $n_f (0 \leq n_f \leq n-1)$ に対して，各々10000 回収束時間を導出し，その収束時間の平均をグラフ化させたものが図 8 である．図 8 は，x 軸に RNG 故障ノード数 n_f，y 軸に収束時間を表したグラフである．図 8 より，RNG 故障ノード数が $n-1$ 個のとき収束時間の期待値が最大だと考えられる．

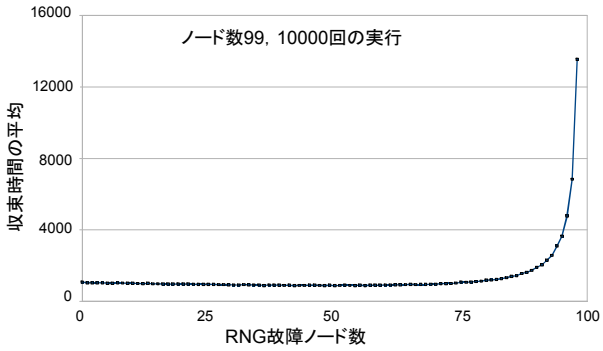

図 8: D_1 の収束時間のグラフ

4.2 巡回時間の評価

定理 2 *RNG* 故障有りモデルにおける *HTCA* の巡回時間の期待値 $T_{cir}(n_f)$ は $2n - n_f$ である．

証明 正常ノードのトークンの処理について考える．正しい状況 $x \in C$ において，正常ノードがトークンを所有しているとき，状況 $f(x)$ でトークンが移動する確率は $\frac{1}{2}$ であり，移動しない確率は $\frac{1}{2}$ である．よって，正常ノードにおいて，トークンが移動するまでに要する期待ラウンド数は $1/(\frac{1}{2}) = 2$ である．次に，RNG 故障ノードのトークンの処理について考える．正しい状況 $x \in C$ において，RNG 故障ノードがトークンを所有しているとき，状況 $f(x)$ でトークンが移動する確率は 1 であり，移動しない確率は 0 である．よって，トークンが移動するまでに要する期待ラウンド数は 1 である．以上より，$T_{cir}(n_f) = 2(n - n_f) + n_f = 2n - n_f$ である．□

5 まとめ

本稿では、乱数生成器が故障する RNG 故障という新たな故障の概念を提案し，RNG 故障有りモデルにおいて，HTCA の評価を行なった．具体的には，まず，RNG 故障ノード数が $n-1$ の RNG 故障有りモデルにおいて，HTCA の収束時間の期待値が $\frac{n(n+1)^2}{4}\lceil \log n \rceil$ であることを示した．また，シミュレーションを用いて，RNG 故障ノード数が $n-1$ のときの収束時間の期待値が最大であることを示した．次に，RNG 故障ノード数が i の RNG 故障有りモデルにおいて，HTCA の巡回時間の期待値が $2n - i$ であることを示した．

今後の課題としては，RNG 故障ノード数が任意の i のときの収束時間の期待値を解析することが考えられる．また，RNG 故障ノードは確率 1 でトークンを巡回させるとしたが，確率 p でトークンを巡回させるときの解析も行いたい．

謝辞 本研究の一部は，文部科学省グローバル COE プログラムの研究助成，日本学生振興会科学研究費補助金 (基盤研究 (B) 20300012，基盤研究 (B) 22300009) によるものである．

参考文献

[1] Edsger Wybe Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, Vol. 17, No. 11, p. 644, 1974.

[2] Ted Herman. Probabilistic self-stabilization. *Information Processing Letters*, Vol. 35, No. 2, pp. 63–67, 1990.

[3] Colette Johnen. Service time optimal self-stabilizing token circulation protocol on anonymous unidirectional rings. pp. 80–91, 2002.

[4] Hirotsugu Kakugawa and Masafumi Yamashita. Uniform and self-stabilizing token rings allowing unfair daemon. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 8, pp. 154–163, 1997.

[5] Hirotsugu Kakugawa and Masafumi Yamashita. Uniform and self-stabilizing fair mutual exclusionon unidirectional rings under unfair distributed daemon. *Journal of Parallel and Distributed Computing*, Vol. 62, No. 5, pp. 885–898, 2002.

[6] Xiangdong Yu and Moti Yung. Agent rendezvous: A dynamic symmetry-breaking problem. *Automata, Languages and Programming*, pp. 610–621, 1996.

[7] Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, Vol. 21, No. 3, pp. 183–199, 2008.