

博 士 論 文

耐故障性能を有する二次元トラス・ネットワークの
適応ルーティングアルゴリズムに関する研究

学 籍 番 号 14T2501 (平成 29. 3. 31 所定単位取得退学)

氏 名 中尾 司ピエール

提 出 月 日 平成 30 年 9 月

指 導 教 員 三浦 康之

湘南工科大学 大学院 工学研究科
博士後期課程 電気情報工学専攻

目次

第 1 章 緒言.....	3
1.1. 背景	3
1.2. 本論文の目的.....	4
1.3. 本論文の構成.....	5
第 2 章 二次元トラス網における適応ルーティング	6
2.1. 二次元トラス網.....	6
2.2. トラス網の適応ルーティング.....	9
2.2.1. ターンモデル	9
2.2.2. North-First(NF)法.....	9
2.2.3. トラス網におけるターンモデルの適用	10
第 3 章 NSF-IP 法.....	12
3.1. North-South First 法.....	12
3.1.1. 定義.....	13
3.1.2. ルーティング・アルゴリズム.....	14
3.2. ルーティング・アルゴリズムの改善	20
3.2.1. ルーティング・アルゴリズム.....	20
3.2.2. デッドロックの回避	21
3.3. 耐混雑性の評価	24
3.3.1. 評価方法	24
3.3.2. Uniform Traffic Pattern	24
3.3.3. Matrix Transpose	25
3.4. 耐故障性の評価	26
3.4.1. 任意 PE の故障による評価.....	26
3.4.2. PE がランダムに故障する場合	28
第 4 章 NSF-FT 法.....	30
4.1. アルゴリズム.....	30
4.2. NSF-FT によるデッドロックの回避	33
4.3. 耐混雑性の評価	33
4.4. 耐故障性評価.....	35
4.4.1. 評価方法	35
4.4.2. 中央 4 箇所故障時.....	35
4.4.3. 隅 4 箇所故障時	36

4.4.4. ランダムに複数箇所の PE を故障させた場合	36
第 5 章 まとめ.....	41
謝辞.....	43
参考文献	44
研究業績	47

第1章 緒言

1.1. 背景

並列処理の分野において、相互結合網に関する研究は、重要なトピックの一つに位置づけられている。 k -ary n -cube 等の直接網により Processing Element (PE) 同士が結合された並列計算機が数多く開発され、商用に提供されている。また一つのチップ上に複数の PE を配置して並列処理を行う「ネットワーク・オン・チップ (NoC)」の分野においては、PE 間を結合する相互結合網の役割はますます大きなものとなっている。そのような背景から、これまでに様々な並列計算機向け相互結合網が提案されている。中でも二次元トーラス網は一般的な相互結合網の一種であり、他の階層型相互結合網の一部として用いられるなど、重要な役割を有している。

相互結合網のルーティングには、経路が固定される固定ルーティングと、途中経路の故障や混雑に応じて経路を適応的に変化させる適応型ルーティング^{[1]~[8][10]~[17]}は、大きく分けて二つの種類に分けられ、それぞれ決まった数の仮想チャンネルが必要となる。後者は前者に比べて、耐故障性が優れ、局所的な混雑に対する耐性が高いことから、さまざまな研究がなされている。Ngai らは、適応型カット・スルー方式を前提としてパケットを制御する方法を提案した^[1]。ただし、この枠組みはハードウェアリソースが豊富に使用できる環境を前提としたものであり、NoC には適さない。Dally らは、固定ルーティング本来の経路に加えて、自由に経路選択可能なバイパス用のチャンネルを設けることで、柔軟性の高い適応ルーティングを行う方法を提案した^{[6][7]}。類似したアプローチによる手法が、Duato の研究においてもなされており、Duato の必要十分条件と呼ばれる理論的枠組とともに紹介されている^[11]。また、結合網の経路長と同等の仮想チャンネルを用意することによってデッドロックを回避する手法が提案されている^[8]。しかしながらこれらの手法は、仮想チャンネルの追加が必要であることから、ハードウェアコストの観点から NoC への実装が適さないことも考えられる。これらの手法を応用し、階層構造を有する複雑な形状のネットワークに適した適応ルーティング・アルゴリズムが提案されている^{[3]~[5]}。これらは、各々の結合網に固有の手法である。二次元トーラスにおいては、固定ルーティングとして、 x 方向、 y 方向、またはその逆の順序で座標を合わせる次元順ルーティング (Dimension Order Routing, DOR)^[9]が用いられる。 k -ary n -cube 向けの手法としては、仮想チャンネルを L, W, H の三種類に分けて、それぞれにおいて適応ルーティングを行う Yang の手法が提案されている^[10]。ただしこの手法は、 L, W, H 各チャンネルをまたぐ適応ルーティングには不適である。 k -ary n -cube の各次元向けにサブネットワークを設ける適応ルーティング法が提案されている^[12]ものの、やはり追加の仮想チャンネルを必要とする手法である。二次元メッシュ網において、各ノードが他の各ノードへの遅延を予測して、転送先を決定する手法が提案されている^[13]。本手法は、二次元メッシュ網向けの手法であり、トーラス網への適用に関しては言及され

ていない。そのため、ハードウェア量を大きく増やせないようなケースでは、追加の仮想チャンネル⁽¹⁸⁾を必要としない手法が求められる。主にメッシュ網向けの適応型ルーティング・アルゴリズムとして、追加の仮想チャンネルを必要としないターンモデルによる方法があり、ターンモデルに基づいたいくつかの手法が提案されている^{[14]~[17]}。Glass らによる、メッシュ網向けのターンモデルによるルーティング・アルゴリズム^{[14][15]}に基づいて、メッシュ網向けの耐故障性を有するルーティング・アルゴリズムの提案^[16]や、ランダムな構造を持つ結合網に適したアルゴリズム⁽¹⁷⁾が提案されている。しかしながら、これらの手法を二次元トラスにそのまま適用する場合、ルーティングに伴う制約条件が非常に大きなものとなる。過去に提案されたターンモデルの一部変更によりトラス網に適用可能な適応型ルーティングがあれば、仮想チャンネルの追加によるハードウェアコストの増大を招くことなく適応型ルーティングを実現することが可能となる。上記に関して三浦らは、二次元トラス網に適用可能な方法として、ターンモデルによる手法の一つである **North First** 法と **South First** 法を組み合わせた **North-South First** 法を提案し、研究を行っている^{[19]~[21]}。過去の研究において、NSF ルーティングを提案し、実験結果により NSF ルーティングが固定ルーティングに比べて高い動的通信性能を有することを示した^[21]。また、静的特性の評価結果を行い、NSF ルーティングにおいて選択可能な経路の数の計算を行った^[20]。しかしながらこれらにおいては、故障 PE が存在しないことを前提にした、耐混雑性の評価のみを行っていた。これらの研究においては、結合網の混雑回避を目的としたルーティング・アルゴリズムの提案に主眼を置いていたため、耐故障性に関して考慮されていなかったのが実情であった。NSF ルーティングは、ソース・ディスティネーション間の最短経路を通過するルーティング・アルゴリズムであったため、耐混雑性には優れていたものの、耐故障性が優れているか否かは不明であった。ルーティング・アルゴリズムが、ある程度の耐故障性を有していれば、故障せずに使用可能な箇所のみを使用してシステムを動作させることが可能になる。

1.2. 本論文の目的

本論文は、NSF 法の改善法として、NSF-IP および NSF-FT の二種類の手法を提案する。NSF-IP は、NSF 法における最短経路の制約を排除することにより、より自由度を高め、耐故障性を向上させたアルゴリズムである。NSF-FT は、NSF-IP に加えて、仮想チャンネルの移動を許すことによりさらに経路選択の自由度を高めたものである。本論文においては、これらの手法の詳細を説明するとともに、シミュレーションによる動的通信性能評価により、NSF-IP、NSF-FT それぞれについて、耐混雑性と耐故障性の評価を行う。

1.3. 本論文の構成

本論文は次の通りに構成される。まず 2 章では、二次元トラス網における適応ルーティングに関する従来研究を紹介する。3 章では初めに、提案手法の一つである NSF-IP に関して、ルーティングアルゴリズムの説明およびデッドロック回避の証明を行う。次に、NSF-IP に関して、耐混雑性及び耐故障性の評価を行う。4 章では、NSF-IP の改善手法として、NSF-FP のアルゴリズムの詳細説明、デッドロック回避の証明、および性能評価を行う。最後に 5 章で本研究の結論と今後の予定について述べる。

第2章 二次元トーラス網における適応ルーティング

2.1. 二次元トーラス網

図 1 に二次元トーラスの構造を示す。二次元トーラスは $N \times N$ の二次元構造をしており、行方向と列方向の座標を定義した部分をノードと呼ぶ(図 1 参照)。ノードはそれぞれ計算用プロセッサである PE と通信用のルータで構成されている。ノード間は相互に通信を行えるように配線されており、上下左右の端にあるノードがそれぞれ矢印線で繋がれている部分をラップアラウンドリンクと呼ぶ。メッシュに比べて 2 倍の分割帯域幅を持ち、平均ホップ数においても有利であることや、さまざまな並列アルゴリズムとの親和性が高いことなどから過去にさまざまな並列計算機に用いられ、トーラス網を内包した結合網も提案されている。二次元トーラス網の固定ルーティング・アルゴリズムとしては次元順ルーティングが広く用いられている。次元順ルーティングはソース PE から、Y 軸方向のチャンネルのみを使って移動を行い、Y 軸方向の座標を合わせた後に、X 軸方向のチャンネルを使ってディステーション PE に移動する。二次元トーラス網において次元順ルーティングを用いる場合にはデッドロックを回避するために、チャンネル L とチャンネル H の 2 本の仮想チャンネルを必要とする。(ただし、ラップアラウンドリンクに含まれる仮想チャンネルは、ラップアラウンドチャンネルとも呼ばれる。)二次元トーラス網の次元順ルーティングにおける仮想チャンネルの選択の方法は、以下のようになる。

- 1) 最初に、Y 座標方向のルーティングを行う際は、チャンネル L を選択する。
- 2) パケットの先頭がラップアラウンドを通過するとき、チャンネル H に移動する。以降、Y 方向のルーティングが行われている間はチャンネル H が使われる。
- 3) Y 座標のルーティングが終了して X 座標のルーティングに移行する際、これまで使用していたチャンネルに関わらず、チャンネル L に移動する。
- 4) パケットの先頭がラップアラウンドを通過するとき、チャンネル H に移動する。以降、X 方向のルーティングが行われている間はチャンネル H が使われる。

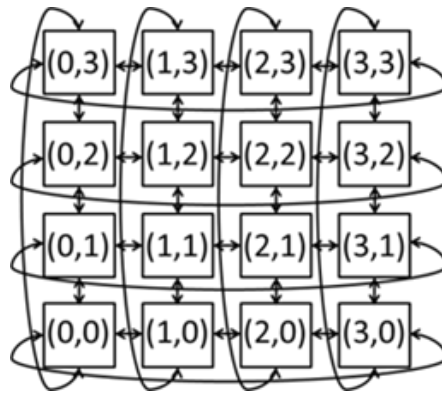


図 1 二次元トーラス網の構造

$N \times N$ トーラスの次元順ルーティングのリンク選択関数とチャンネル選択関数をそれぞれ図 2 と図 3 に示す. ここで, トーラスの各 PE の番号は, 図 1 で示されるように, (x, y) の座標で示される. また, Y 方向のチャンネルを Y+ および Y- と表記し, X 方向のチャンネルを X+ および X- と表記する. リンク選択関数の 4 個の入力は, それぞれ現在ノードの x, y 座標, 目的地ノードの x, y 座標である. 現在地ノード座標と目的地ノード座標について比較した結果 X+, X-, Y+, Y-, 方向のリンクおよびノードへの出力リンクである OUT のいずれかを出力する. チャンネル選択関数の 3 個の入力は, それぞれ現在の方向, 現在のチャンネル, 行き先の方向を示している. “現在の方向” および “行き先の方向” は, X+, X-, Y+, Y- の 4 つの状態を持ち, “現在のチャンネル” は, チャンネル L (L), チャンネル H (H), およびラップアラウンドチャンネル (W) の 3 つの状態を持つ. 出力は, チャンネル L およびチャンネル H の 2 つの状態を持つ. ただし, 選択関数によって選択されたリンク (すなわち転送先のリンク) がラップアラウンドリンクである場合, 2 節にて述べたように, 前述の出力がチャンネル L, チャンネル H のいずれであるかに関わらず, 行き先のチャンネルは無条件でラップアラウンドチャンネルに含まれることになる.


```

// Link Selection Function for Dimension-Order Routing
Link_Select_DOR (cx, cy, dx, dy)
  cx, cy;      // current node   $0 \leq cx, cy \leq N-1$ 
  dx, dy;      // destination   $0 \leq dx, dy \leq N-1$ 
{
  if(cy  $\neq$  dy){          // dimension Y
    dist_y = (N+dy-cy)%N;
    if( $1 \leq$  dist_y  $\leq$  N/2) return Y+;
    else          return Y-;
  }
  else if(cx  $\neq$  dx){     // dimension X
    dist_x = (N+dx-cx)%N;
    if( $1 \leq$  dist_x  $\leq$  N/2) return X+;
    else          return X-;
  }
  else return OUT
};

```

図 2 次元順ルーティングのリンク選択関数

```

// Channel Selection Function for Dimension-Order Routing
Channel_Select_DOR (cd, cc, nd)
    cd;          // current direction      ∈ {Y+, Y-, X+, X-}
    cc;          // current channel ∈ {L, H, W}
    nd;          // next direction   ∈ {Y+, Y-, X+, X-}
    {
        if(cc ∈ L)      return L; // before wrap around traversal
        else if(cc ∈ W) return H; // in wrap around
        else
            if(cd ∈ {X+, X-} & nd ∈ {Y+, Y-})
                return L;          // Y-routing → X-routing
            else
                return H;          // after wrap around traversal
    }

```

図 3 次元順ルーティングのチャンネル選択関数

2.2. トーラス網の適応ルーティング

2.2.1. ターンモデル

ターンモデルはルーティング・アルゴリズムを決定する際に用いるモデルの一種で、本モデルを使用してルーティングにおける進路変更(ターン)するパターンに禁止を加えることにより、パケットのデッドロックを起こさないようにすることができる。二次元トーラス/メッシュの場合、考えられる進路変更(ターン)は全部で8通りあるため、これらの中から禁止を加える。ターンモデルに基づく適応型ルーティング法として、いくつかの手法が提案されている。二次元メッシュ網においては、これらの手法はいずれも、8通りの進路変更パターンのうち2箇所を禁止するというものである。NSF法では主要な手法であるNorth First(NF)法およびSouth First(SF)法に基づいて、二次元トーラス網へ応用している。

2.2.2. North-First(NF)法

図4に、二次元メッシュにおける次元順ルーティングのターンモデルを図5に、NF法のターンモデルを示す。次元順ルーティングは8通りの進路変更パターンのうち4箇所を禁止している。具体的には左方向へ移動した後上方向へ移動するターン(図4の(a)), 右方向に移動した後上方向に移動するターン(図4の(b)), 左方向に移動した後下方向に移動するターン(図4の(c)), 右方向に移動した後下方向に移動するターン(図4の(d))の4箇所を禁止している。

図5に示したNF法では、左方向に移動した後上方向に移動するというターンと(図5の(e)),

右方向に移動した後上方方向に移動するターン(図5の(f))の2箇所のみ禁止を加えたものとなっている。その結果としてY+(North)方向が、必ずルーティング経路の最初に選択されることから、North-First法と名付けられた^[14]。NF法では禁止されるターンの種類が少ないため、経路選択の自由度が高くなっている。また、右方向に移動した後下方方向に移動するターンと、左方向に移動した後下方方向に移動するターンの2箇所のみ禁止を加え、Y-(South)方向が、必ずルーティング経路の最初に選択される South First法も存在する^[14]。

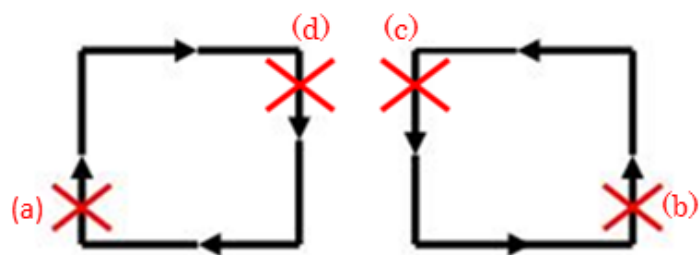


図4 二次元メッシュ網の次元順ルーティングのターンモデル

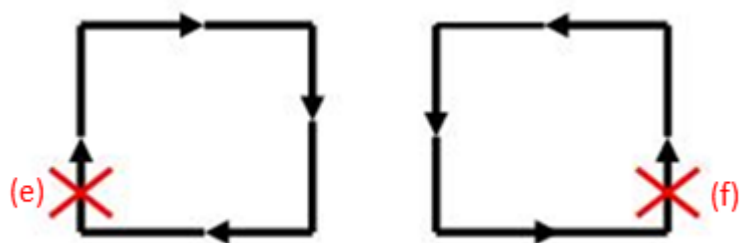


図5 二次元メッシュ網のNorth First(NF)法のターンモデル

2.2.3. トーラス網におけるターンモデルの適用

トーラス網において North First 法などのターンモデルをそのまま用いる場合、以下のような相違点がある。

- (1) トーラス網においてはラップアラウンドチャネルを通過することにより、循環依存によるデッドロックが発生するため、メッシュ網よりも厳しい制限を課す必要がある。
- (2) トーラス網のルーティングには最低 2 本の仮想チャネルを必要とする。このようなル

ルーティング法を採用する場合、各々のチャンネルに対して異なるターンモデルを適用することにより、従来よりも柔軟性の高い適応ルーティングの実現が可能になる。

(1)に関して、トーラス網において North First 法を用いた際に循環依存が発生する例を図 6 に示す。

図 6 は、North First 法に基づき、PacketA から PacketD がそれぞれ宛先ノードに向けてルーティングを行っている様子を図示したものである。図の例ではワームホールルーティング⁹⁾に基づき、各パケットが「フリット」と呼ばれる、1 クロックサイクルに転送できる量のデータに分けられて転送を行っている。図 6 においては、PacketA は PacketB の後尾により進路を妨げられている。同様に、PacketB は PacketC の、PacketC は PacketD の後尾により進路を妨げられている。それぞれのパケットが互いの進路を妨げるため、転送が行われず、結果として循環依存の構造となっている。このような状態をデッドロックと呼ぶ。このような状況が発生した場合、結合網が機能しなくなるため、このような状況が起こらないことが理論的に示される(デッドロックフリーである)ことが求められる。固定ルーティングである「次元順ルーティング」では、図 6 のうち PacketA と PacketC のターンが起こり得ない(それぞれ、PacketA は図 4 の右から下の、PacketC は図 4 の左から下のターンが存在する)ため、このような現象は発生しない。以上のような問題があることから、2 次元トーラス網において適応ルーティングを考える際には、より複雑なターンの制限について考慮する必要がある。(2)に関して、本稿では、channel-L に対して NF 法を、channel-H に対して SF 法をベースとした手法の適用を検討する。

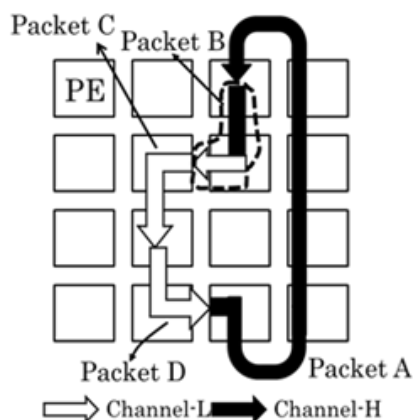


図 6 North First 法をトーラスに用いた時に発生する循環依存

第3章 NSF-IP 法

3.1. North-South First 法

NF 法や SF 法のターンモデルを二次元トラス網にそのまま使用すると、ラップアラウンドチャンネルを通過するパケットにより図 6 のような循環が起こる。そこで、下記の方針に従って、従来の NF 法や SF 法をもとに、付加的な制限を加えることによりデッドロックを回避する。

1) チャンネル-H のルーティングに対して、SF 法を適用する。ただし、パケットがチャンネル-H を経てチャンネル-L に戻るような経路を、当該時点以後に選択すること判明している場合 (Y 方向のラップアラウンドチャンネルを通過後、Y 方向のルーティングを行って X 方向のルーティングに移行する場合、新たに選択される X 方向チャンネルは必ず channel-L でなければならない)、循環が発生するため、適応ルーティングを行わず固定ルーティングを行う。

2) チャンネル-L のルーティングに対して、NF 法に基づいた手法を適用する。チャンネル-L のルーティングにおいては、いったんラップアラウンドチャンネルを通過した上で、チャンネル-H→チャンネル-L の経路が存在する。それにより図 6 に示した循環が発生するため、NF 法にさらにターンの禁止を加える必要がある。そこで、図 7 のように NF 法に更に 1 つ禁止を加えることにより循環を回避する。ターンモデルより、8 つのターンのうち 3 つに制限を加える。具体的には、右方向に移動した後上方向に移動するターン、左方向に移動した後上方向に移動するターン、右方向に移動した後下方向に移動するターンの 3 つに制限を加える。このようなルーティング・アルゴリズムを「制限型 North First 法 (あるいは制限型 NF)」と呼ぶことにする。

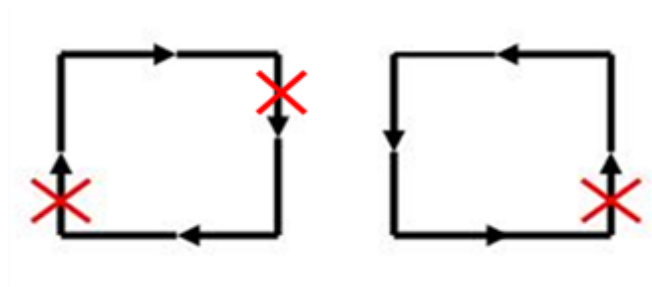


図 7 制限型 North First 法

3.1.1. 定義

以下に NSF の定義について述べる. 一般に, 相互結合網 \mathbf{G} は, $\mathbf{G} = (\mathbf{V}, \mathbf{C})$ と表記される. ここで, \mathbf{V} はノード (頂点または PE とも呼ぶ) の集合, \mathbf{C} はチャンネルの集合である. チャンネルはノードとノードを結ぶので, ノード $v_1 \in \mathbf{V}$ からノード $v_2 \in \mathbf{V}$ へ直接フリットを送るチャンネルを, $c_{v_1 \rightarrow v_2} \in \mathbf{C}$ などと表記する.

これらの各ノードおよびチャンネルに対して個々を識別するための記号を付与する.

定義 1 :

$N \times N$ ノードを持つ二次元トーラス網

$\mathbf{G}_{2DT_N} = (\mathbf{V}_{2DT_N}, \mathbf{C}_{2DT_N})$ における各ノード $v \in \mathbf{V}_{2DT_N}$ の記号を, 網中の縦横方向の座標から, 以下のように定義する.

$$v = (x, y) \quad \text{ただし, } x = 0, 1, \dots, N-1, \quad y = 0, 1, \dots, N-1$$

すなわち, \mathbf{V}_{2DT_N} は, 以下のように定義される.

$$\mathbf{V}_{2DT_N} = \{v = (x, y) \mid x \in [0, N-1], y \in [0, N-1]\} \quad (1)$$

定義 2 :

$N \times N$ ノードを持つ二次元トーラス網

$\mathbf{G}_{2DT_N} = (\mathbf{V}_{2DT_N}, \mathbf{C}_{2DT_N})$ における各チャンネル $c \in \mathbf{C}_{2DT_N}$ を, 次元 $d \in (X, Y)$, 方向 $\delta \in (+, -)$, チャンネル種別 $ch \in (L, W, H)$ により, 以下のようにラベル付けする.

$$c = (d\delta, ch)$$

なお, ここで, X: X 方向, Y: Y 方向, L: チャンネル L, W: ラップアラウンドチャンネル, H: チャンネル H を意味する. すなわち, \mathbf{C}_{2DT_N} は, 以下のように定義される.

$$\mathbf{C}_{2DT_N} = \{c = (d\delta, ch) \mid d \in (X, Y), \delta \in (+, -), ch \in (L, W, H)\} \quad (2)$$

この場合, 複数のチャンネルに対して同じラベルが付与される場合がある. 以後の説明において, $(d+, ch)$ と $(d-, ch)$ が一組で表記される場合がある. その場合, $(d\pm, ch)$ などと表記する. また, 必要に応じて, Y 座標が n であるチャンネル $(d\delta, ch)$ を, チャンネル $(d\delta, ch)_n$, X 座標が n であるチャンネル $(d\delta, ch)$ を, チャンネル $(d\delta, ch)^n$ などと表記する.

定義 3 :

二次元トラス網 G_{2DT_N} におけるソースノードを $n_{src} \in V_{2DT_N}$, デスティネーションノードを $n_{dst} \in V_{2DT_N}$ とする. あるルーティング・アルゴリズム R を用いて経路が選択され, その繰り返しにより, n_{src} から n_{dst} へパケットが送られる. また, パケットが通過し得るチャンネルの集合を, $P_{n_{src} \rightarrow n_{dst}}^R \subset C_{2DT_N}$ とする.

3.1.2. ルーティング・アルゴリズム

提案手法は, channel-L で制限型 NF 法, channel-H で SF 法に基づくルーティングを行うものである. 制限型 NF 法と SF 法はそれぞれ, チャンネル(Y-, L), チャンネル(Y+, H) を使用するパケットのみが適応ルーティングの対象となることから, ここでは, チャンネル(Y+, ch) を使用するパケットと使用しないパケットに分けて検討する. そのため, ルーティングの過程において特定のチャンネルを使用するか否かに関する以下の記述を定義する.

定義 4 :

2 節において定義された次元順ルーティングアルゴリズム R_{DOR} , および任意のチャンネルの集合 $C_n \subset C_{2DT_N}$ において,

$$\exists p \in C_n, p \in P_{n_{src} \rightarrow n_{dst}}^{R_{DOR}} \quad (3)$$

を満たす場合, $Use(n_{src}, n_{dst}, C_n, R_{DOR})$ を満たすと定義する. すなわち, n_{src} から n_{dst} へ向かうパケットが, 次元順ルーティングの過程において, あるチャンネルの集合 C_n のいずれかを通過する場合, $Use(n_{src}, n_{dst}, C_n, R_{DOR})$ を満たすと定義する.

定義 5 :

2 節において定義された次元順ルーティングアルゴリズム R_{DOR} および任意のチャンネルの集合 $C_n \subset C_{2DT_N}$ において,

$$\forall p \in C_n, p \notin P_{n_{src} \rightarrow n_{dst}}^{R_{DOR}} \quad (4)$$

を満たす場合, $\overline{Use}(n_{src}, n_{dst}, C_n, R_{DOR})$ を満たすと定義する. すなわち, n_{src} から n_{dst} へ向かうパケットが, 次元順ルーティングの過程において, あるチャンネルの集合 C_n を一切使用

しない場合, $\overline{\text{Use}}(n_{src}, n_{dst}, \mathbf{C}_n, R_{DOR})$ を満たすと定義する.

定義 6 :

特定の $d_n \in (X, Y)$, $\delta_n \in (+, -)$, $ch_n \in (L, W, H)$ において, チャンネル $\mathbf{C} = (d_n \delta_n, ch_n)$ の集合を, $\mathbf{C}_{d_n \delta_n, ch_n}$ と定義する. すなわち

$$\mathbf{C}_{d_n \delta_n, ch_n} = \{c = (d_n \delta_n, ch_n)\} \equiv \{\mathbf{C} \in \mathbf{C}_{2DT_N}\} \quad (5)$$

とする.

定義 7 :

特定の $d_n \in (X, Y)$, $ch_n \in (L, W, H)$ において, チャンネル $(d_n +, ch_n)$ および $(d_n -, ch_n)$ の和集合を, $\mathbf{C}_{d_n \pm, ch_n}$ と定義する. すなわち

$$\mathbf{C}_{d_n \pm, ch_n} \equiv \mathbf{C}_{d_n +, ch_n} \cup \mathbf{C}_{d_n -, ch_n} \quad (6)$$

とする.

定義 8 :

特定の $d_n \in (X, Y)$, $\delta_n \in (+, -)$ において, チャンネル $(d_n \delta_n, ch)$ の集合を, $\mathbf{C}_{d_n \delta_n}$ と定義する. すなわち

$$\mathbf{C}_{d_n \delta_n} \equiv \mathbf{C}_{d_n \delta_n, L} \cup \mathbf{C}_{d_n \delta_n, W} \cup \mathbf{C}_{d_n \delta_n, H} \quad (7)$$

とする.

提案手法のリンク選択関数およびチャンネル選択関数を, それぞれ図 8, 図 9 に示す. リンク選択関数は, 図 2 の固定ルーティングと同様, $X+$, $X-$, $Y+$, $Y-$, および, ノードへの出力リンクである「OUT」のいずれかのリンクを出力する. ただし, 図 2 の 4 個の入力

の他に、「現在のチャンネル」を示す入力が必要とする。

提案手法では、 $\text{Use}(n_{src}, n_{dst}, C_{Y+}, R_{DOR})$ を満たす場合と $\overline{\text{Use}}(n_{src}, n_{dst}, C_{Y+}, R_{DOR})$ を満たす場合で異なるポリシーによるチャンネル選択が行われる。前者では、「現在以後ラップアラウンドチャンネルを通過する可能性がない」場合にのみ SF 法による適応ルーティングが行われ、それ以外のケースでは固定ルーティングとなる。後者では、ソースノードからルーティングが始まり、最初にラップアラウンドチャンネルに到達する（あるいは目的地ノードに到達する）までは制限型 NF 法による適応ルーティングが行われる。

図 8 のアルゴリズムでは、はじめに①において、X 方向、Y 方向それぞれについて、ラップアラウンドリンクを使用するか否かを判別する。この場合、ソース・ディスティネーション PE の X、Y 各方向の座標をもとに、以下のように判別する。

- チャンネル(Y+,C_{Y+})を使用しない場合、③の処理を行う。チャンネル H における SF 法は固定ルーティングと等価になるため、チャンネル L においてのみ制限型 NF 法に基づく適応ルーティングを行う。その際、通過するラップアラウンドチャンネルを、(C_Y, W), (X±, W) の順番に維持するため、現在地からディスティネーション PE までに (C_Y, W) を通過する予定がない場合、または (C_{X±}, W) を跨がない場合のみ制限型 NF 法を行い、それ以外の場合には固定ルーティングを行うこととしている。
- カレント PE とディスティネーション PE の X 座標の差が、N/2 以下の場合、X 方向のラップアラウンドチャンネルを跨がないので、h_wrap=0 とし、そうでない場合を h_wrap=1 とする。
- 同様に、カレント PE とディスティネーション PE の Y 座標の差が、N/2 以下の場合、v_wrap=0 とし、そうでない場合を v_wrap=1 とする。

次に、Y+方向のチャンネル（チャンネル(Y+,ch)）を使用するか否かによって以下のように経路を選択する。

- $\text{Use}(n_{src}, n_{dst}, C_{Y+}, R_{DOR})$ を満たす場合、②の処理を実行する。この場合、チャンネル L における制限型 NF 法は固定ルーティングと等価になるため、チャンネル H においてのみ SF 法に基づく適応ルーティングを行う。このとき、現在地からディスティネーション PE までに X 方向、Y 方向いずれのラップアラウンドチャンネルを使用しない場合 ($\overline{\text{Use}}(n_{src}, n_{dst}, C_{X±,W}, R_{DOR})$ かつ $\overline{\text{Use}}(n_{src}, n_{dst}, C_{Y±,W}, R_{DOR})$ の場合)、パケットをチャンネル H に送ってルーティングを続けることができるので、SF 法に基づいた適応ルーティングを実施する。それ以外の場合で、チャンネル H が使用される可能性があるのは、縦方向のラップアラウンドチャンネル (Y+, W) の通過直後で、かつ今後横方向のラップアラウンドチャンネル (X±, W) を通過する予定がある場合 ($\text{Use}(n_{src}, n_{dst}, C_{X±,W}, R_{DOR})$ を満たす)のみである。実際、この場合においても SF 法に基づいた適応ルーティングが可能と考えられるが、後述のデッドロック・フリーの証明が困難なため、本稿の手法では X 方向のルーティングのみを行い、先に (X±, W) を通過した後に適応ルーティングを行うものとした。それ以外の場合には、チャンネル L を使用するため、固定ルーティン

グを適用する.

- $\text{Use}(n_{src}, n_{dst}, C_{Y+}, R_{DOR})$ を満たす場合, ③の処理を行う. チャンネル H における SF 法は固定ルーティングと等価になるため, チャンネル L においてのみ制限型 NF 法に基づく適応ルーティングを行う. その際, 通過するラップアラウンドチャンネルを, $(Y-, W)$, $(X\pm, W)$ の順番に維持するため, 現在地からディスティネーション PE までに $(Y-, W)$ を通過する予定がない(現在のチャンネルを $ch \in C_{2DT_N}$ としたときに $\overline{\text{Use}}(n_{src}, n_{dst}, C_{Y-}, R_{DOR})$ を満たす)場合, または $(X\pm, W)$ を跨ぐ前のみ制限型 NF 法を行い, それ以外の場合には固定ルーティングを行うこととしている. チャンネル選択関数の入力, 図 3 における固定ルーティングのチャンネル選択関数の 3 個の入力に加えて, 現在地ノードと目的地ノードの x, y 座標を示す 4 個の入力を必要とする. 新たな 4 個の入力により, 「現在以後ラップアラウンドチャンネルを通過する可能性がない」か否かの判定を行う. 判定結果をもとに, 「現在以後ラップアラウンドチャンネルを通過する可能性がない」かつ「 $\text{Use}(n_{src}, n_{dst}, C_{Y+}, R_{DOR})$ を満たす」場合のみ, H チャンネルを選択し, それ以外のケースでは固定ルーティングと同様の処理を行う. 固定ルーティングの場合と同様, 出力はチャンネル L およびチャンネル H の 2 つの状態を持つが, 選択された先のチャンネルがラップアラウンドリンクである場合, 出力は無条件でチャンネル W となる.

```

// Link Selection Function for NSF Algorithm
Out_DOR=Link_Select_DOR(cx,cy,dx,dy);
Link_Select_Prop (cx, cy, cc, dx, dy)
  cx, cy;          // current node   $0 \leq cx, cy \leq N-1$ 
  cc;              // current channel  $\in \{L, H, W\}$ 
  dx, dy; // destination   $0 \leq dx, dy \leq N-1$ 
{
  if(dx-cx  $\geq N/2$ )   h_wrap = 1; ①
  else                h_wrap = 0;
  if(dy-cy  $\geq N/2$ )   v_wrap = 1;
  else                v_wrap = 0;

  dist_x = (N+dx-cy)%N;
  dist_y = (N+dy-cy)%N;
  if(1  $\leq$  dist_y  $\leq$  N/2) // Y+ direction ②
  {
    if(h_wrap=0 & v_wrap=0)
      return adaptive_SF(cx, dx);
    else if(h_wrap=1 & v_wrap=0)
      return Link_Select_DOR(cx, cy, dx, cy);
    else if((h_wrap=1 & v_wrap=1)
      &(((1  $\leq$  dist_x  $\leq$  N/2) & (cx=N-1))
      or((dist_x > N/2) & (cx = 0))))
      return Y+;
    else
      return DOR(cx, cy, dx, dy);
  }
  else if(cy  $\neq$  dy) // Y- direction
  {
    if(cc=0)          return out_NF; ③
    else              return out_DOR;
  }
  else if(cx  $\neq$  dx) return out_x;
  else               return OUT;
}

out_SF = adaptive_SF(cx, dx){ //adaptive routing of SF algorithm
  if(cx=dx)          return Y+;
  else if(buffer_is_full(Y+, H)=TRUE)
    return out_x;
  else
    return Y+;
}

out_NF = adaptive_NF(cx, dx){ //adaptive routing of NF algorithm
  dist_x = (N+dx-cx)%N;
  if((cx=dx)or((1  $\leq$  dist_x  $\leq$  N/2) & (cx=N-1))
  or((dist_x < N/2) & (cx=0)))
    return Y+;
  else if(N/2 < dist_x) // X- direction
    return out_x;
  else if(buffer_is_full(Y-, L)=TRUE) // X+ direction
    return X+;
  else
    return Y-;
}

out_x = x_route(cx, dx){
  dist_x = (N+dx-cx)%N;
  if(1  $\leq$  dist_x  $\leq$  N/2)
    return X+;
  else
    return X-;
}

```

図 8 NSF 法のリンク選択関数

```

// Channel Selection Function for NSF Algorithm
Out_ch_DOR = Channel_select_DOR(cd, cc, nd);
Channel_Select (cx, cy, dx, dy, cd, cc, nd)
  cx, cy;      // current node  $0 \leq cx, cy \leq N-1$ 
  dx, dy;      // destination  $0 \leq dx, dy \leq N-1$ 
  cd;          // current direction  $\in \{Y+, Y-, X+, X-\}$ 
  cc;          // current channel  $\in \{L, H, W\}$ 
  nd;          // next direction  $\in \{Y+, Y-, X+, X-\}$ 
{
  if(dx-cx $\geq$ N/2)          h_wrap = 1;
  else                      h_wrap = 0;
  if(dy-cy $\geq$ N/2)          v_wrap = 1;
  else                      v_wrap = 0;

  dist_y = (N+dy-cy)%N;
  if((1 $\leq$ dist_y  $\leq$ N/2) // Y+ direction
    & (h_wrap=0 & v_wrap=0)) return H;
  else                      // Others
    return Out_ch_DOR;
}

```

図 9 NSF 法のチャンネル選択関数

3.2. ルーティング・アルゴリズムの改善

3.2.1. ルーティング・アルゴリズム

前節のアルゴリズムは、トラス網の一部（「ソースノードからラップアラウンドリンク」「ラップアラウンドリンクからディスティネーションノード」「ソースノードからディスティネーションノード」などの、チャンネル L およびチャンネル H のみで構成されている、実質的にはメッシュ網の構造となっている部分）のルーティングの途中において、パケットが迂回経路を通らないアルゴリズムである。このようなアルゴリズムを若干変更して、迂回経路を通ることを可能にすることにより、パケットが故障個所を避けて通ることが可能となる。

図 10 に NSF と提案手法のルーティング経路の相違を示す。NSF においては、図 10 の経路①や②のように、現在地ノードから、目標ノードまでの距離が縮まる方向へのルーティングのみ行われるが、提案手法では、図 10 の経路③や④のように、いったん目標ノードまでの距離が遠ざかる方向へのルーティングが許される。

そこで、NSF 法の耐故障性の改善のために、ルーティング・アルゴリズムの改善を行った。従来の NSF 法においては最短経路をルーティング・アルゴリズムにおいて保証していたが、本稿では保証していたソース・ディスティネーション間の最短経路を無くすことにより耐故障性能の確保をはかった。図 11 に改善されたルーティングアルゴリズムの一部を示す。改善はリンク選択関数を改良することで実現している。アルゴリズムは図 8 で示した NSF 法の adaptive_SF 関数で行っていた処理に変更を加えた。改善前との大きな違いは、 $(X+,H)$ と $(X-,H)$ を選択する適応ルーティングを行っている点である。これにより、最短経路か否かによらず混雑や故障を避けたルーティングが可能となる。

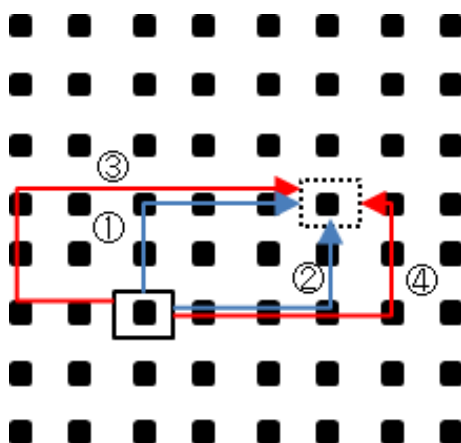


図 10 NSF 法と NFS-IP 法によるルーティング経路の違い

```

// Link Selection Function for NSF-IP Algorithm
adaptive_SF(cx, dx){
    if(buffer_is_full(Y+,H)=FALSE)
        return Y+;
    else if(cx==0)
        return X+;
    else if(cx==N-1)
        return X-;
    else{
        dist_x = (N+dx-cx)%N;
        if(1<= dist_x <= N/2){
            if(buffer_is_full(X+,H)=FALSE)
                return X+;
            else
                return X-;
        }
        else {
            if(buffer_is_full(X-,H)=FALSE)
                return X-;
            else
                return X+;
        }
    }
}

```

図 11 NSF-IP 法のリンク選択関数

3.2.2. デッドロックの回避

前節で示されたルーティング・アルゴリズムがデッドロックを生じないことを証明するためには、チャンネルをノードとし、ルーティング・アルゴリズムに従ってチャンネル間で直接的なパケットの送受信が生じる可能性のあるノード（チャンネル）同士を矢印で結んだ有向グラフである「チャンネル依存グラフ」を作成し、各チャンネルに番号をつけ、チャンネルにつけた番号が、チャンネル依存グラフの矢印の方向に向かって必ず昇順（または降順）になることが証明される必要がある⁽¹¹⁾⁽²²⁾⁽²³⁾。チャンネル番号が昇順（降順）になるということは、チャンネル同士が順序関係を持つことになるため、チャンネルが循環依存を起こさないためである。Turn model に基づくルーティング・アルゴリズムにおいては、各 PE からの出力チャンネルに対して、PE アドレスに基づいて番号付けを行う方法が一般的である。前述のように、2次元トーラス網の場合、2個の仮想チャンネルを持つため、 $N \times N$ トーラスの4リンク $\times 2$ チャンネル=8本のチャンネルに対して、以下のような4次元のチャンネル番号 CN をつける。

$$CN(x, y, d, ch) = (g_m, c_1, g_s, c_2) \tag{1}$$

ここで、 x ($0 \leq x \leq N - 1$) および y ($0 \leq y \leq N - 1$) は、PE の x, y 座標、 $d \in$

$\{Y+, Y-, X+, X-\}$ は、チャンネルの方向、 $ch \in \{L, H, W\}$ は、チャンネルの種類である。また、 g_m, c_1, g_s, c_2 をそれぞれ主グループ、第一座標、副グループ、第二座標と名付ける。これらは、それぞれ以下の基準で番号付けを行う。

- 主グループ g_m

d や ch をもとに、チャンネルの順序関係を設定する。順序関係に基づいて、表のようにグループ分けして g_m の値とする。

表 1 d および ch による g_m の値

d	ch	g_m
Y+	L, W	0
Y-, X-	L, W	1
Y-	H	
X+	L, W	2
X+, X-, Y+	H	3

- 第一座標 c_1

c_1 は、 g_m の値に基づいて表 2 のように与えられる。

表 2 g_m による c_1 の値

g_m	c_1
0	y
1	$N - x$
2	0
3	y

- 副グループ g_s

副グループ g_s は、同一の g_m 内において、 d および ch に基づいてチャンネルの順序関係を規定するものである。 $g_m = 0$ および $g_m = 2$ においては、単一の d 、 ch のみを持つため、 $g_s = 0$ のみを持つ。 $g_m = 1$ および $g_m = 3$ においては、表 3 のように g_s を決定する。

表 3 g_s の値

g_m	d	ch	g_s
1	Y-	L, W	0
	Y-	H	1
	X-	L, W	2
3	X+, X-	H	0
	Y+	H	1

- 第二座標 c_2

第二座標 c_2 は、同一の d や ch において順序関係を定める値で、 $d = Y+, Y-, X+,$ および $X-$ のチャンネルに対してそれぞれ $c_2 = y, N - y, x,$ および $N - x$ と定める。

以上をまとめると、各チャンネルにおけるチャンネル番号は図 12 のようになる。このようにチャンネル番号を設定すると、ルーティング経路に従ってチャンネル番号が必ず昇順になるため、デッドロックの発生を防ぐことができる⁽²⁰⁾⁽²¹⁾。

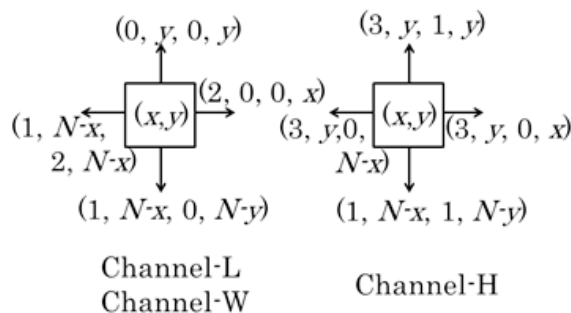


図 12 チャンネル番号

3.3. 耐混雑性の評価

3.3.1. 評価方法

ワームホールルーティングを再現するソフトウェアシミュレータを用いて、256 PE を持つ 16×16 2-D トーラス網における動的通信性能を評価する。評価に用いるシミュレータはルータ回路と相互接続網のハードウェア構造を機能ユニットレベルで忠実に再現したもので、C 言語で書かれている⁽²⁵⁾。このシミュレータは各 PE で 1 サイクルごとにある条件でパケットを発生し、ランダムに選択した他の PE に送信する⁽²⁶⁾。動的通信性能は、DOR と NSF(North South First ルーティング)と NSF-IP(改善 North South First ルーティング)により評価される。評価のための通信パターンは、Uniform, Matrix Transpose の 2 パターンである。パケットのサイズを 16 フリットとし、シミュレーション時間 T を、 $T=50000$ とする。各物理リンクの仮想チャネルの数を 2 本とし、各チャネルのバッファの量を 8 フリットとした。

3.3.2. Uniform Traffic Pattern

Uniform Traffic Pattern は送信先が均等にランダムに決定される通信である。相互結合網の動的通信性能は、スループットと平均転送時間により特徴づけられる。転送時間は、先頭のフリットがネットワークに入った時刻から、最後尾のフリットがディスティネーション PE に到着するまでの時刻と定義づけられる。平均転送時間は、全パケットの転送時間の平均値である。スループットは、単位時間あたりに目的地 PE に到着するフリットの数の、時間あたりの平均値である。パケットの送信要求確率を r とし、 T サイクルの間シミュレーションを実行し、その間に到着したパケットの数と転送時間を記録する。平均転送時間とスループットを計算し、前者を縦軸、後者を横軸としてグラフにプロットする。図 13 にネットワークのスループットに対する平均転送時間を示す。グラフの横軸がスループット、縦軸が平均転送時間である。図 13 に示すように、NSF-IP 法は NSF 法と同程度に DOR と比較してスループットが高いことが明らかになった。また、NSF-IP 法と NSF 法を比較した場合スループットはほぼ同等であり改良による大幅な性能低下は見られなかった。ランダム通信のように結合網が全体的に混雑するような通信パターンでは、Turn model のような付加的な仮想チャネルを用いない方法では、適応ルーティングにより混雑を避ける効果が限定的となる。しかしながら本手法の場合、一部のパケット (Y+を使用し、かつラップアラウンドチャネルを通らないパケット) を直接チャネル H に送り込んでいるため、チャネルに対する負荷が分散される。そのため、スループットが若干向上しているものと思われる。

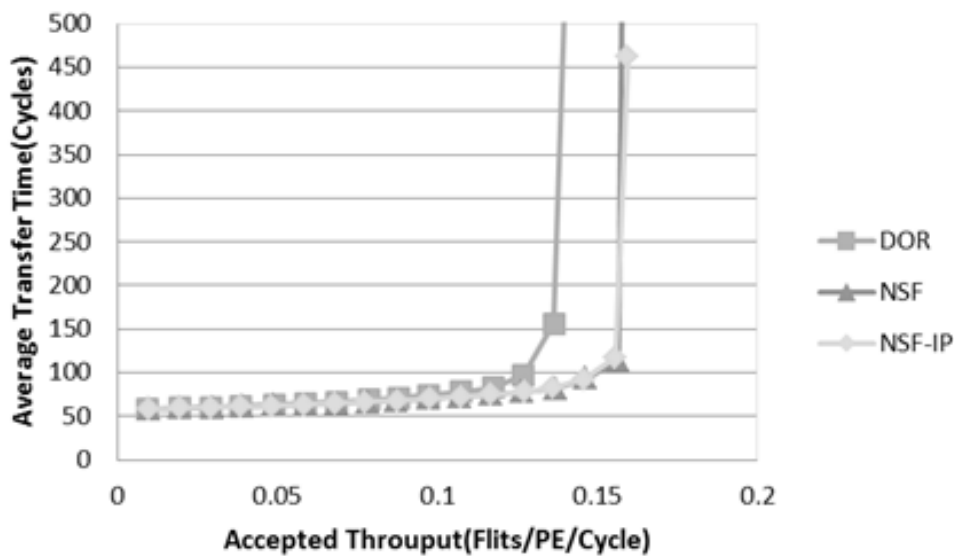


図 13 Uniform Traffic の結果

3.3.3. Matrix Transpose

Matrix Transpose は行列転置に基づいた通信方法であり，対角線をまたいで折り返す転送方法である． 8×8 のメッシュ／トーラスにおける Matrix Transpose の通信パターンを図 14 に示す．Matrix Transpose においては，図 14 に示すように，結合網の中央付近に通信の負荷が集中する傾向がある．PE と同じ数のデータがあると仮定し，行列 $A = \{i, j\}$ の各要素について，PE $\{x, y\}$ に a_{xy} を割り当て，行列転置の通信を行った．そのため Matrix Transpose による通信パターンは，PE (x, y) から PE (y, x) への通信となる．この通信について上記の通信を 10 回行ったものと 50 回行ったものについてシミュレーションを行い終了までにかかるサイクル数の測定を行った．表 4 に実験結果を示す．

表 4 Matrix-Transpose の結果

loops	NSF-IP	NSF	DOR
10	2482	2559	2910
50	12425	12389	13773

表 4 は 1 列目がループ数を示しており 2 列目以降は 3 手法についてのそれぞれの終了までのサイクル数の値となっている．ループ 10 回，ループ 50 回ともに DOR と NSF-IP を比較すると NSF-IP のほうがサイクル数が低く DOR より性能が出ていることがよみとれる．また，NSF-IP を NSF と比較した場合は同程度の性能であると分かり，改良による大幅な

性能低下は見られない。Matrix Transpose のように負荷に局所性のある通信パターンにおいても、適応ルーティングの効果によりネットワークの負荷が改善することが分かった。

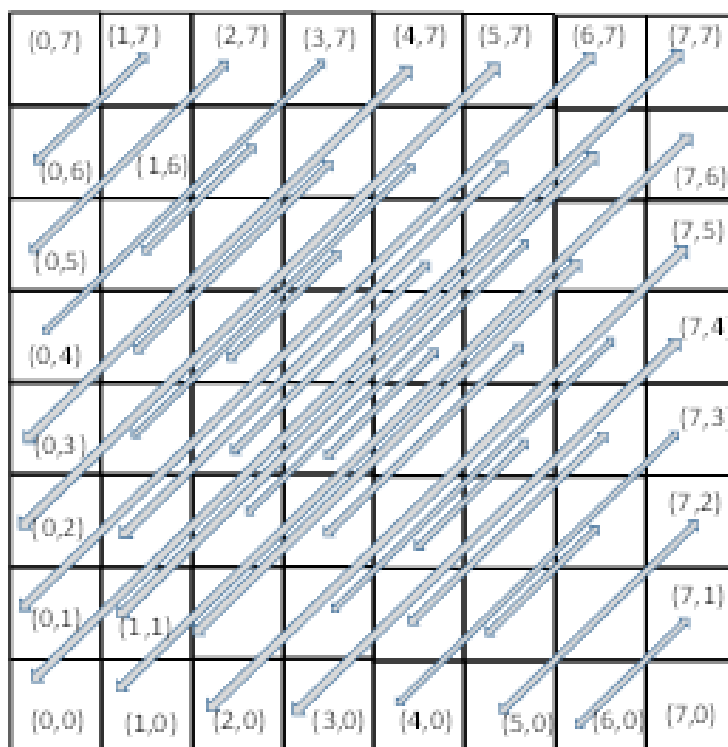


図 14 Matrix Transpose の通信パターン

3.4. 耐故障性の評価

3.4.1. 任意 PE の故障による評価

256PE を持つ 16×16 2-D トーラス網シミュレータ内に任意の箇所に故障状態の PE を作成し、通信を行うことにより耐故障性の評価を行う。始めに、256 個の PE のうち一つの PE を故障 PE と仮定した。通信方法はそれぞれの PE がランダムに 1 個ずつパケットを各 PE 1 対 1 に対応する形で通信を行う通信を 1 回とし、1 回あたり 255 個のパケットが送信される。以上の方法によって DOR, NSF, NSF-IP, 3 つのアルゴリズムについてパケットの不着数を測定し、それぞれループ 1, 3, 5 回において 10 回の結果の平均をとり、それぞれのアルゴリズムについてのパケットの平均不着数を集計したものを表 5 に示す。

表 5 に示すように、改良案である NSF-IP は NSF よりも不着数が低く耐故障性能が改善す

ることが明らかになった。DOR と比較した場合においても NSF-IP は耐故障性能が確保されているとみられる。

表 5 一つの PE を故障 PE とした場合の不着パケット数

Method	loop1	loop3	loop5
NSF-IP	5.2	85.3	448.3
NSF	6.2	102.1	493.7
DOR	9.2	136.7	574.0

次に、256 個の PE を持つ 16×16 2-D トーラス網内のラップアラウンドリンクを持つ 4 隅の PE を故障 PE とし、DOR、NSF、NSF-IP、3 つのアルゴリズムについてパケットの不着数を測定し、それぞれループ 1、3、5 回において 10 回の結果の平均をとり、パケットの平均不着数を集計したものを表 6 に示す。NSF-IP と DOR、NSF-IP と NSF の結果をそれぞれ比較した場合、ラップアラウンドリンクを含む 4 隅の故障という形においては提案手法がラップアラウンドリンクを含む隅に対す耐故障性を向上するアルゴリズムでないため NSF と比較した場合、若干の不着数増加となった。

表 6 4 隅の PE を故障 PE とした場合の不着パケット数

Method	loop1	loop3	loop5
NSF-IP	15.9	213.4	698.2
NSF	15.7	210.4	693.4
DOR	19.2	248.6	740.9

次に、256 PE を持つ 16×16 2-D トーラス網内の中央 4 箇所 PE を故障 PE とし、DOR、NSF、NSF-IP、3 つのアルゴリズムについてパケットの不着数を測定し、それぞれループ 1、3、5 回において 10 回の結果の平均をとり、パケットの平均不着数を集計したものを表 7 に示す。NSF-IP と DOR、NSF-IP と NSF の結果をそれぞれ比較した場合、中央 4 箇所故障という形においても NSF-IP が耐故障性の性能が高いことがあきらかとなった。また、先の 4 隅の故障 PE を作成した実験の結果(表 6 参照)と比較した場合、中央 4 箇所はラップアラウンドリンクを含まないため 3 手法それぞれの不着パケット数が減少したとみられる。

表 7 中央 4 箇所の PE を故障 PE とした場合の不着パケット数

Method	loop1	loop3	loop5
NSF-IP	14.7	188.3	648.8
NSF	16.8	204.9	681.2
DOR	21.1	251.5	742.4

3.4.2. PE がランダムに故障する場合

256 PE を持つ 16×16 2-D トーラス網内にランダムに 1, 2, 4, 8, 16 個の PE を故障している PE とし, DOR, NSF, NSF-IP, 3つのアルゴリズムについてパケットの不着数を測定し, それぞれループ 1, 3, 5 回において 10 回の結果の平均をとり, パケットの平均不着数をプロットしたものをループ 1, 3, 5, について図 15, 図 16, 図 17 に示す. グラフの縦軸が平均不着数, 横軸が PE の故障数である.

図 15, 図 16, 図 17 すべてに見受けられる点として, 横軸の故障数の増加にしたがって, 不着数も漸増することがわかる. 図 15 提案手法と DOR について 1 箇所の故障と 4 箇所の故障で差がみられるのは故障 PE の選定を完全にランダムにしているためラップアラウンドリンクを含む PE が故障箇所として多く選定されてしまった点が影響を受けていると考えられる. また, ループ 1 においてはネットワーク内を流れるパケットの総数が少ないためこのような結果となったと考えられる. それ以外においては NSF-IP が DOR, NSF と比較して不着数が少ないことが明らかとなっており, NSF-IP の実装により耐故障性能が向上しているとみられる.

このような結果になった理由として表 6 と表 7, また図 15, 図 16, 図 17 の故障 PE=4 の時の結果を比較した場合から以下のことが考えられる.

- 中央の故障(表 7 参照)より四隅の故障(表 6 参照)の不着数が多い. これは, 隅に関してはラップアラウンドリンクが付近にあり, ラップアラウンドリンクは経路選択の制約が大きいため故障による影響が大きいためである.
- ランダムに故障箇所を選定したほうが不着パケットが多い(図 15, 図 16, 図 17 参照)こともあきらかとなった. この原因は, 不着による影響が広がるためである. このようなことを低減するためには, 故障 PE を矩形領域でまとめて故障領域で扱うことが有効であると知られているため, いかにして故障領域を生成するかが今後の課題となる.

図 17 については不着パケットが大半を占めるため差が出にくくなっている. その場合においても NSF や NSF-IP は, 幾分不着のパケットを減らしている.

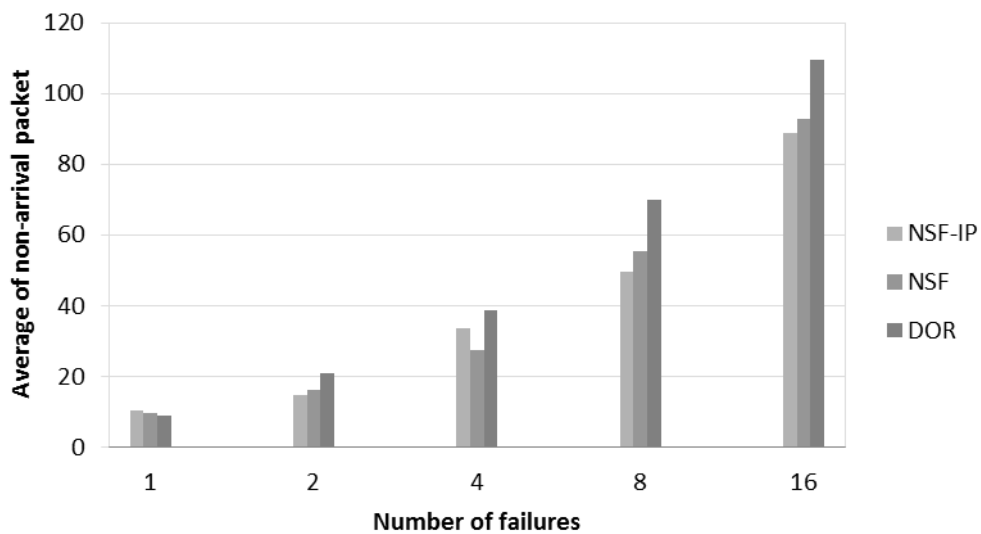


図 15 ループ 1 回の場合においてランダムに 1, 2, 4, 8, 16 ヶ所故障する場合の不着
パケット数

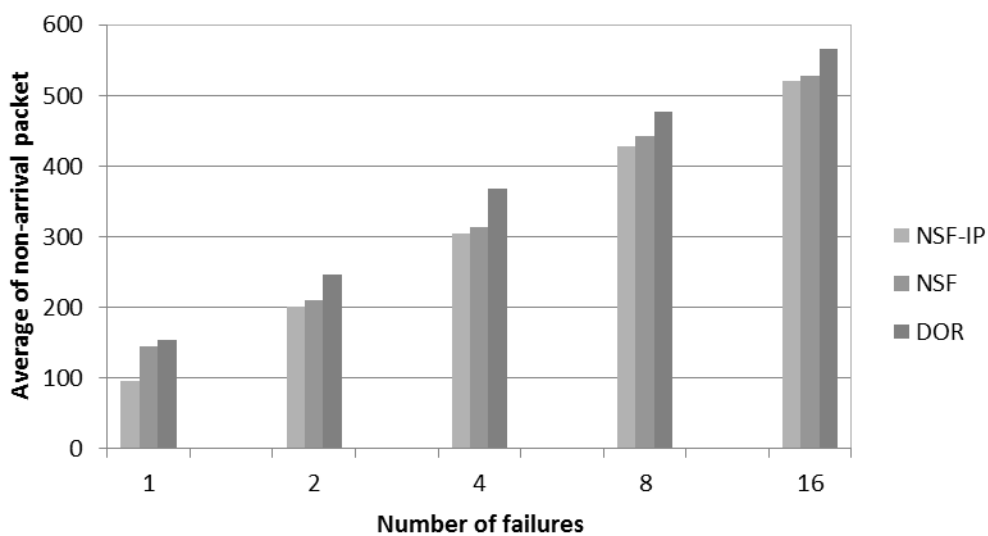


図 16 ループ 3 回の場合においてランダムに 1, 2, 4, 8, 16 ヶ所故障する場合の不着
パケット数

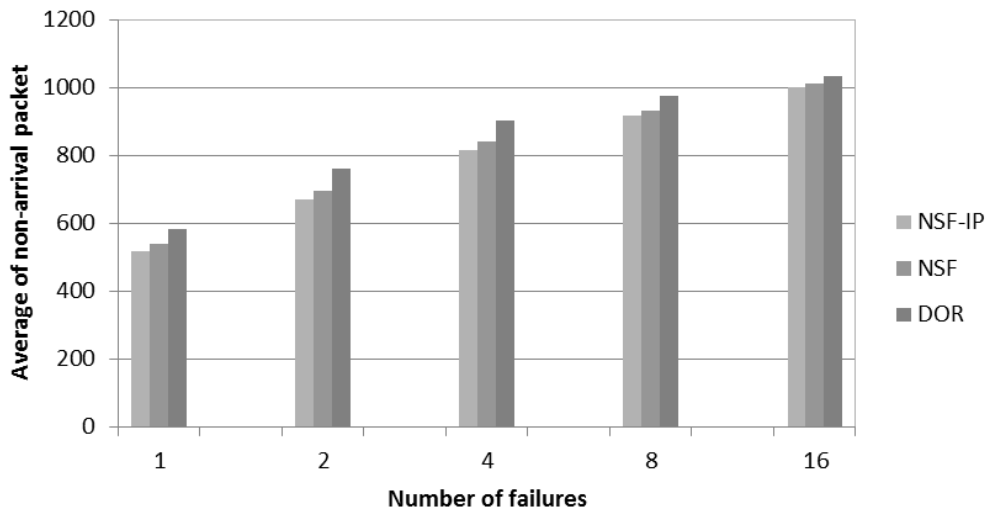


図 17 ループ 5 回の場合においてランダムに 1, 2, 4, 8, 16 ヶ所故障する場合の不着
パケット数

第4章 NSF-FT 法

4.1. アルゴリズム

NSF-FT 法は、NSF-IP 法における channel-L 上のパケットに対して、下記の条件を満たす場合においてパケットの先頭を channel-H に移動させることにより、ルーティング先の選択枝を増やし、NSF-IP において実現できなかった耐故障性の向上を目指したものである。NSF-FT による故障個所の回避の例を図 18 に示す。図 18 は、パケットがソース PE からデスティネーション PE へ転送される状況を示している。

従来手法では、まず channel-L を利用し経路①のような転送が行われる。図のような故障個所が存在する場合、 α 点において故障個所によってパケットの進路が塞がれるため、デスティネーション PE へ到達することはできなくなる。そこで、次のホップ先が故障状態である場合は、パケットの先頭を強制的に channel-H へ移動させる。これにより SF 法に基づく適応ルーティングが可能となる。結果として、図のような場合はパケットの進路が変わり、channel-H を用いて経路②のような進路が選ばれる。結果としてパケットは故障個所を回避することができる。

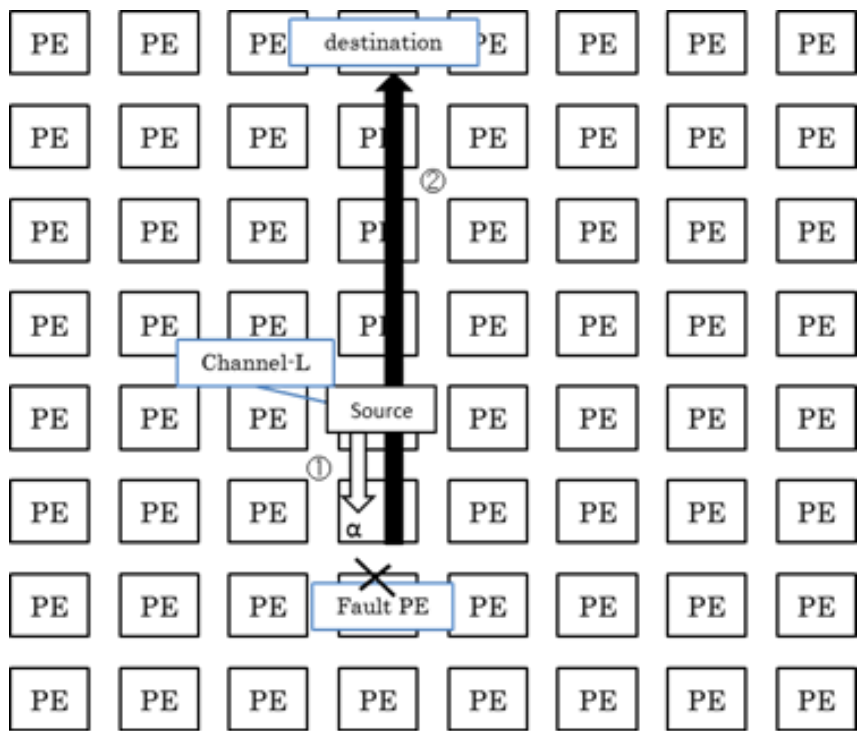


図 18 想定される故障パターン

図 19 に NSF-FT アルゴリズムのリンク選択関数を示す. 図の「Out_NSF_IP」は NSF_IP によるリンク選択関数の選択結果, 「Out_NSF_IP_H」は, channel-H において, SF アルゴリズムによりルーティングを行う場合のリンク選択関数の選択結果である. また, status(Out_NSF_IP)は, NSF_IP によるリンク選択先の PE の状態を返しており, PE が故障している場合は fault の値を返すものとしている. 図のアルゴリズムでは, まず NSF_IP によるリンク選択先の PE の確認を行い, 当該 PE が故障していない場合は, NSF_IP によるリンク選択関数の選択結果を選ぶものとしている. 当該 PE が故障していた場合は, SF アルゴリズムによるルーティングを行う.

図 20 に, NSF-FT のチャンネル選択関数を示す. 図の「Ch_NSF_IP」は, NSF_IP によるチャンネル選択関数の選択結果を示している. 図のチャンネル選択関数では, まず NSF_IP によるリンク選択先の PE の確認を行い, 当該 PE が故障していない場合は, NSF_IP によるチャンネル選択関数の選択結果を選ぶものとしている. 当該 PE が故障していた場合は, channel-H を選択するものとしている.


```

// The link Selection Function of the NSF-FT Algorithm.
Link_Select_NSF_FT(cx, cy, cc, dx, dy)
{
    Out_NSF_IP = Link_Select_NSF_IP (cx, cy, cc, dx, dy);
    if(dy>cy) Out_NSF_IP_H = adaptive_SF(cx, dx)
    else      Out_NSF_IP_H = DOR(cx, cy, cc, dx, dy);

    if(status(Out_NSF_IP) == fault)
        return Out_NSF_IP_H;
    else
        return Out_NSF_IP;
}

```

図 19 NSF-FT のリンク選択関数

```

// The Channel Selection Function of the NSF-FT Algorithm.
Channel_Select_NSF_FT(cx, cy, cc, dx, dy)
{
    Out_NSF_IP = Link_Select_NSF_IP(cx, cy, cc, dx, dy);
    Ch_NSF_IP = Channel_Select(cx, cy, dx, dy, cd, cc, nd);

    if(status(Out_NSF_IP)==fault)
        return H;
    else
        return Ch_NSF_IP;
}

```

図 20 NSF-FT のチャンネル選択関数

4.2. NSF-FT によるデッドロックの回避

NSF-FT は、channel-L と channel-W からチャンネル H に移動する手法である。図 12 によると、channel-L から channel-H への移動において、明らかにチャンネル番号が降順になるのは、 $C_{X+,L}$ または $C_{X+,W}$ から $C_{Y-,H}$ に移動するケースにおいてのみである。このようなケースにおいては、チャンネル番号が $(2, 0, 0, X)$ から $(1, N-x, 1, N-y)$ と変化するため、チャンネル番号が降順になる。そこで、まずはこのような順序逆転が起こらないことを示す。

$C_{X+,L}$ または $C_{X+,W}$ から $C_{Y-,H}$ に移動するケースとして考えられるパターンは

- ① $C_{X+,L}$ から $C_{Y-,L}$ を経由して $C_{Y-,H}$ に移動する場合
- ② $C_{X+,L}$ から $C_{X+,W}$ 、 $C_{X+,H}$ を経由して $C_{Y-,H}$ に移動する場合
- ③ $C_{X+,L}$ または $C_{X+,W}$ の経路中、進路に故障 PE が存在するためにパケットが channel-H に移動して、 $C_{Y-,H}$ が使用される場合の三種類である。

図 7 の Turn モデルより、channel-L における制限型 NF 法においては、右方向から下方向へのターンは禁止されていることから、 $C_{X+,L}$ から $C_{Y-,L}$ への遷移は起こらないため、①のような遷移は起こらない。また、channel-H における SF 法においても、右方向から下方向へのターンは禁止されているため、 $C_{X+,H}$ から $C_{Y-,H}$ への遷移は起こらない。したがって、②のような遷移は起こらない。上記の理由から、channel-L、channel-W、channel-H のいずれにおいても右方向から下方向へのターンは禁止されている。そのため、右下方向にディスタネーション PE が位置するルーティングにおいては、先に Y 方向の座標をそろえてから C_{X+} を使用することになる。故に、 $C_{X+,L}$ または $C_{X+,W}$ が使用される場合、その後に C_{Y-} が使用されることはないので、③のような遷移は起こらない。以上から、 $C_{X+,L}$ または $C_{X+,W}$ から $C_{Y\pm}$ へのターンは起こらない。

なお、 $C_{X-,L}$ または $C_{X-,W}$ から $C_{Y-,H}$ に移動するケースは、 $C_{Y-,H}$ に属するチャンネルの方が副グループ g_s の値が小さくなる $((1, N-x, 2, N-x)$ から $(1, N-x, 1, N-y)$ へと変化するため、一見すると順序関係の逆転が起こる可能性があるように見えるが、 $C_{X-,L}$ を通過する時点で第一座標 c_1 が変化して値が増加するため、順序関係の逆転は起こらない。

4.3. 耐混雑性の評価

ワームホールルーティングを再現するシミュレータを用いて 256PE をもつ 16×16 二次元トラス網における動的通信性能を評価する。評価に用いるシミュレータはルータ回

路と相互結合網のハードウェア構造をクロスバスイッチ、FIFO、マルチプレクサおよびデマルチプレクサ、制御ユニット、プロセッサコアなどの機能ユニットレベルで関数化することによって忠実に再現し、C 言語で記述されている[26][27]. このシミュレータは、各 PE のプロセッサコアで 1 サイクルごとに、ある条件でパケットを生成すし、サイクルレベルのシミュレーションを実行する. 生成されたパケットは、サイクルが変わるごとに (ハードウェア構造を忠実に再現ながら) シミュレータ中の各機能ユニット間の移動を繰り返し、ランダムに選択した他の PE のプロセッサコアへと送られる. このシミュレータを用いて動的通信性能を、DOR, NSF, NSF-IP, NSF-FT について評価する. 評価に用いる通信パターンは Uniform パターンである. 相互結合網における動的通信性能はスループットと平均転送時間により特徴づけられる. 転送時間については、先頭のフリットがネットワークに入った時刻から、最後尾のフリットが目的地の PE に到達するまでの時刻と定義づけられる. 平均転送時間は全パケットの転送時間の平均値である. パケットの送信要求確率を r として、 T サイクルの間シミュレーションを実行し、その間に到着したパケットの数と転送時間を記録する. 平均転送時間とスループットを計算し、前者を縦軸、後者を横軸としてグラフにプロットする. パケットのサイズを 16 フリットとし、シミュレーション時間 T を、 $T=50000$ とした. 各物理リンクの仮想チャンネルの数を 2 本とし、各チャンネルのバッファの量を 8 フリットとした. 評価に用いる Uniform Traffic Pattern は送信先が均等にランダムに決定される通信である. 図 21 にネットワークのスループットに対する平均転送時間を示す. 図 21 に示すように NSF-FT 法は NSF-IP 法, NSF 法と同程度に DOR と比較してスループットが高いことが明らかとなった. NSF-FT 法と NSF-IP 法を比較した場合においてもスループットはほぼ同等であり提案手法の実装による性能の低下は見られなかった.

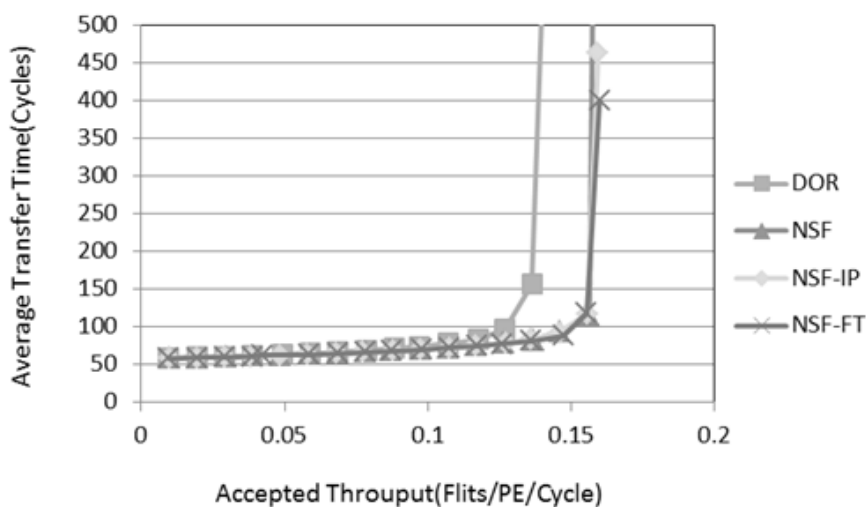


図 21 Uniform Traffic の結果

4.4. 耐故障性評価

4.4.1. 評価方法

256 個の PE を持つ 16×16 二次元トラス網シミュレータ内に故障状態の PE を作成し、通信を行うことにより耐故障性能の評価を行った。耐故障性能の評価に用いた通信方法は、それぞれの PE がランダムに 1 個ずつパケットを各 PE1 対 1 に対応する形で通信を行う通信を 1 回の通信とする。このため、故障 PE が存在しない場合は、1 回あたり 256 個のパケットがシミュレータ内に送信されることとなる。また、シミュレータ内において故障ノードにぶつかったパケットはその場にとどまる条件としている。このため、その場で動かなくなったパケットは削除されずそのまま残ることになる。以上の環境において DOR, NSF, NSF-IP, NSF-FT, 4 つのアルゴリズムについて様々ないくつかの故障パターンを作成しパケットの不着数を測定する。不着となるパケットの数を測定することにより、「提案手法によって、ディスティネーション PE に到着しないパケットをいかに減らすことができるか」を評価することが可能となる。本稿の提案手法によって完全なパケット到達性を保障することはできないものの、不着となるパケットの数を少なくすることにより、不着パケットの破棄と再送に伴うオーバーヘッドを抑えることができる他、他の手法[28]との組み合わせによる完全なパケット到達性の保障に繋げることが可能となる。不着数は故障個所を除いた到着すべきパケット総数から到着したパケット総数を引いて求めている。不着数の測定に際しては、1 回分の通信を 1 ループ数とし 1, 3, 5 ループにて測定する。上記の各々について 10 回の測定を行い平均をとったものを平均不着数として表に示す。

4.4.2. 中央 4 箇所故障時

256PE のうち中央の 4 箇所を故障 PE として測定した結果を表 11 に示す。4 箇所を故障個所としているため、ループ 1 回あたり PE 間で送られるパケットは 252 パケットとなる。以上を踏まえループ 1 に注目して各種法について比較すると、DOR と比較して NSF, NSF-IP, NSF-FT の 3 手法は不着率が低く耐故障性が優れていることが分かる。また、前節 NSF, NSF-IP, NSF-FT の 3 手法で不着数を比較した場合、提案手法である NSF-FT は不着数が NSF-IP と比較してループ 1 では増加している。NSF-FT がループ 1 において NSF-IP に比べて増加が見られる理由は、4.1 節の図において述べられるような方法により故障個所を回避するため、結果としてルーティング経路が複雑になったためであると考えられる。ループ 3 からループ 5 について比較をした場合においては不着率が低く改善が見られた。また、4 手法それぞれについて、ループ数 1 とループ数 3 の結果を比較した場合、ループ数 3 の結果はループ数 1 の結果の 3 倍より大きな値となる。これは、故障ノードに妨害されたパケットに、別のパケットが進路を妨害された結果である。その結果、大量の

未着パケットが結合網中に残るためこのような結果になったと考えられる。NSF-FT と NSF-IP が DOR や NSF に比べて改善が見られる理由は、最短経路の制約を取り除いたため、これらの手法に比べて経路の選択肢が多くなったためと考えられる。

表 11 中央 4 箇所故障時の測定結果

手法	ループ数 1	ループ数 3	ループ数 5
NSF-FT	14.8	179.0	639.8
NSF-IP	14.7	188.3	648.8
NSF	16.8	204.9	681.2
DOR	21.1	251.5	742.4

4.4.3. 隅 4 箇所が故障時

ラップアラウンドリンクを含む隅 4 箇所の PE を故障 PE とし、平均不着数を測定したものを表 12 に示す。ループ数 1 からループ数 5 の全てにおいて提案手法が優れていた。NSF-FT は NSF-IP と比較して、NSF-IP では改善しなかった点が改善している。故障の少ない経路にパケットが逃げているからである。これは、本来ラップアラウンドリンクを使う経路であったはずパケットが使わずにディスティネーション PE へパケットが向かうためである。結果として隅においての耐故障性の効果が NSF-IP より大きいことが明らかとなった。

表 12 四隅故障時のパケットの不着数

手法	ループ数 1	ループ数 3	ループ数 5
NSF-FT	13.1	182.8	652.5
NSF-IP	15.9	213.4	698.2
NSF	15.7	210.4	693.4
DOR	19.2	248.6	740.9

4.4.4. ランダムに複数箇所の PE を故障させた場合

256 個の PE を持つ 16×16 二次元トーラス網シミュレータ内に、ランダムに 1, 2, 4, 8, 16 個の PE を故障している PE と仮定し、DOR, NSF, NSF-IP, NSF-FT 4つのアルゴリズムについて、パケット不着数を測定した。それぞれループ数 1, 3, 5 回において 10 回の結果平均をとり、パケットの平均不着数をプロットしたものをループ数 1, 3, 5, にそれぞれの結果を図 22, 図 23, 図 24 に示す。グラフの縦軸は平均不着数である。また

NSF-FT の平均不着数について他の手法と比較した割合を表 13 に示す。表 13 の「DOR→FT」の覧の値は、 $(\text{NSF-FT における不着パケット数})/(\text{DOR における不着パケット数})$ である。同様に「IP→FT」「NSF→FT」はそれぞれ、 $(\text{NSF-FT における不着パケット数})/(\text{NSF-IP における不着パケット数})$ 及び、 $(\text{NSF-FT における不着パケット数})/(\text{NSF における不着パケット数})$ である。表中の網掛け部は NSF-FT による耐故障性の向上が見られなかった箇所である。

この実験において各ループ数において要した平均遅延時間について図 25, 図 26, 図 27 に示す。各故障数に応じて転送にかかったサイクル数を縦軸としてプロットしたものである。これらのうち、図 22 においてはばらつきが多く見られた。これは、図 22 の条件においては計測対象となるパケットの数が少ないためと考えられる。他に、ばらつき傾向が見受けられる原因として NSF-FT はパケットが channel-H へ他の手法より多く流れ込むため、結果として channel-H の経路が混雑してしまいサイクル数の増加や不着数の増加を招いたと考えられる。図 23 のループ数 3 の条件においては NSF-FT を実装したことによる不着数の改善がすべての手法において見られた。これは表 13 の結果からも合わせて分かる。図 24 においては全ての故障箇所において NSF-FT の実装による効果が見られた。これは表 13 に示した変化率からも分かる通りである。以上より、適度に混雑している状況において NSF-FT の耐故障性の効果が大きくなる傾向が見られた。図 25～27 より、平均遅延時間は DOR 及び NSF-FT がやや長いことが分かる。DOR の平均遅延時間が長い理由は、パケットの進路に関する選択肢がすくないため、ネットワーク中のパケットの待ち時間が長いと考えられる。NSF-FT の平均遅延時間が長くなる理由は、転送経路自体が長くなるためと考えられる。4 節の図 19 において述べられるように、NSF-FT は故障 PE によってリンク選択のルールを大きく変える手法のため、転送経路自体が長くなる傾向がある。

このことから、中程度以上の混雑状況と故障数と両方がある状況下において提案手法の効果が高まる可能性があることがあきらかとなった。これは、NSF-FT が混雑と PE 故障の双方を経路選択の判定に含んでいるためであると考えられる。

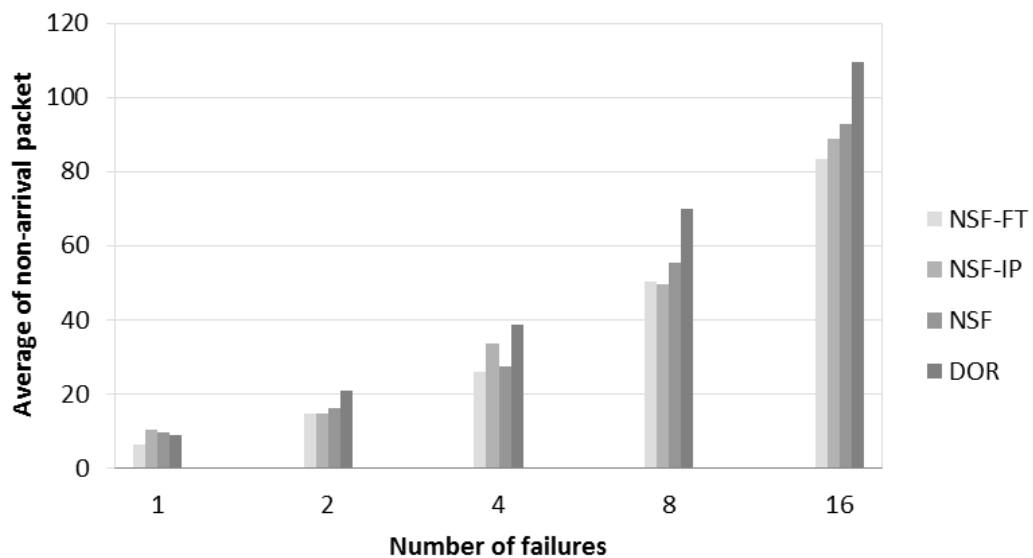


図 22 ループ数 1 における平均不着数

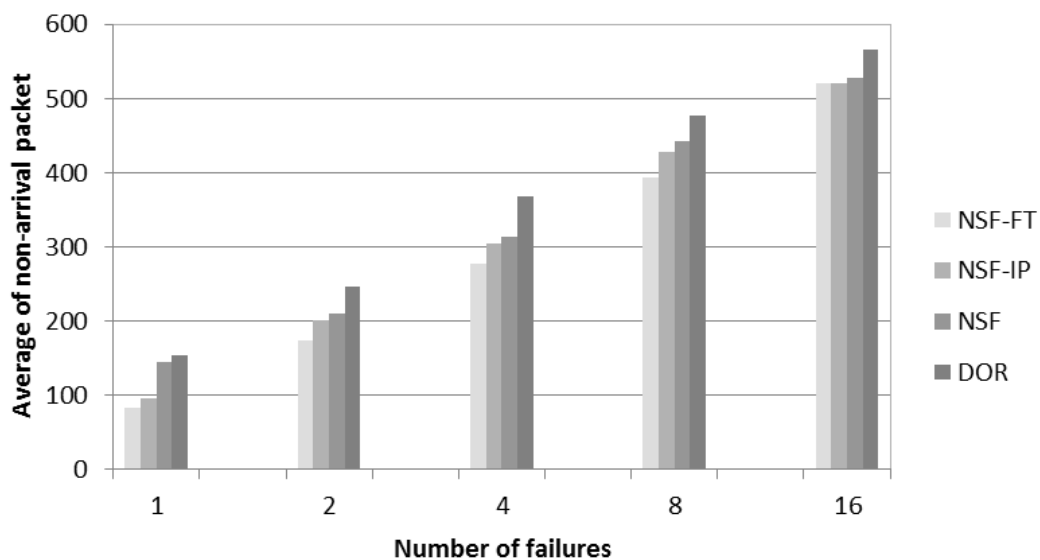


図 23 ループ数 3 における平均不着数

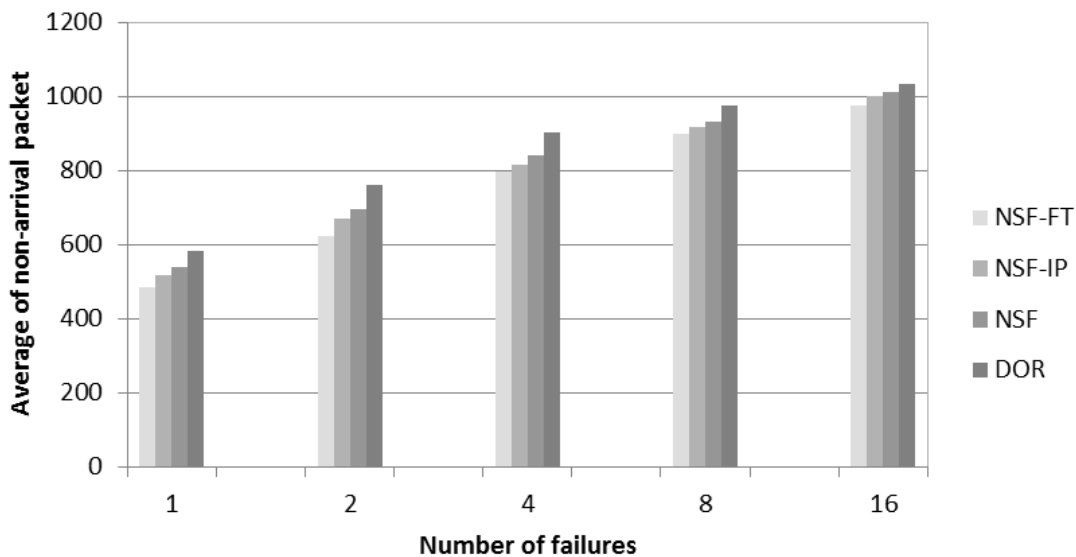


図 24 ループ数 5 における平均不着数

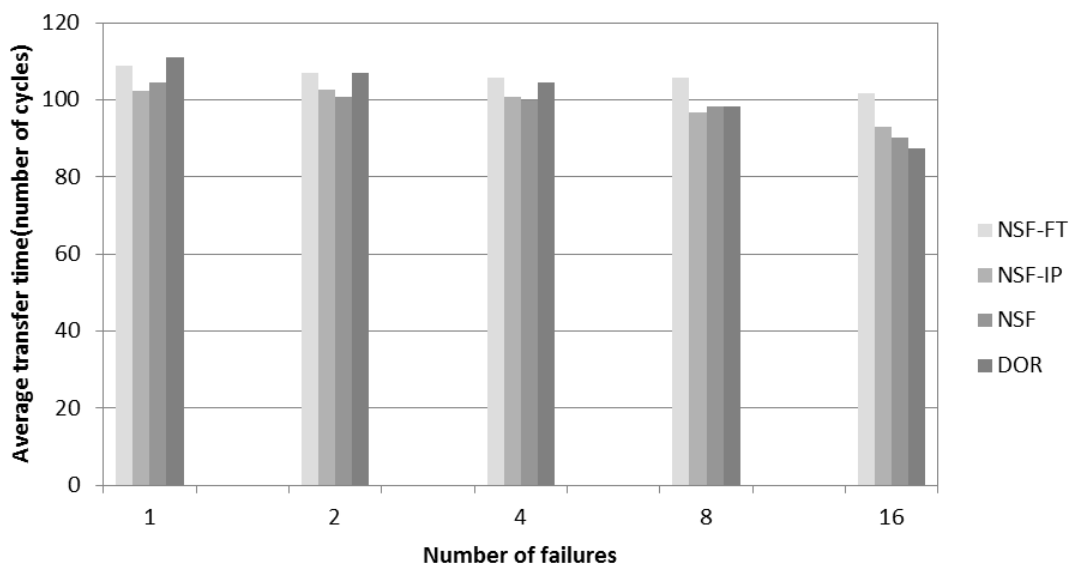


図 25 ループ数 1 における平均遅延時間

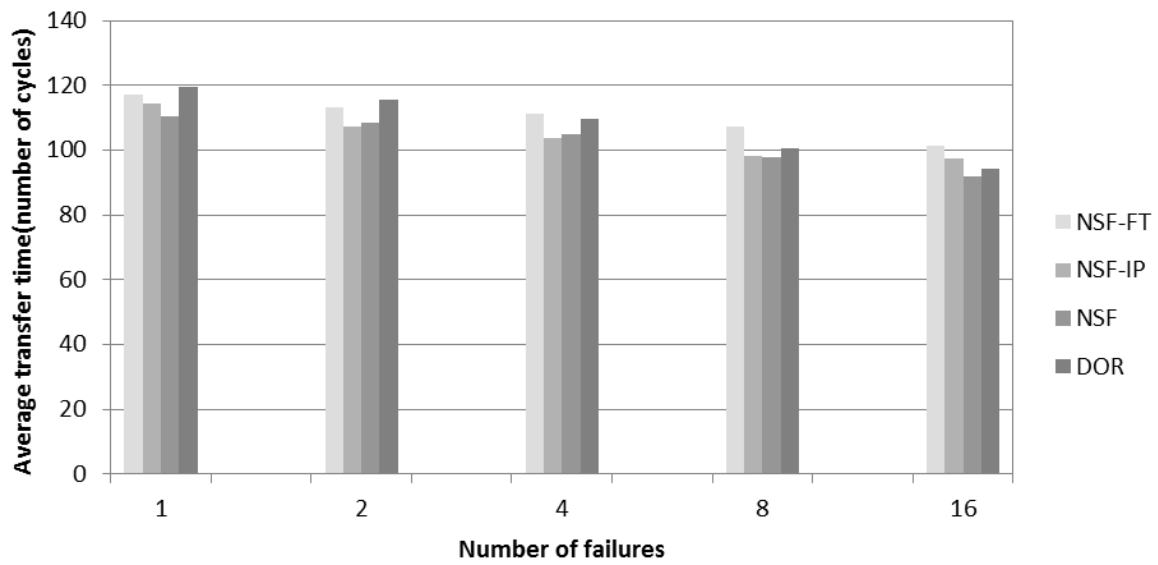


図 26 ループ数 3 における平均遅延時間

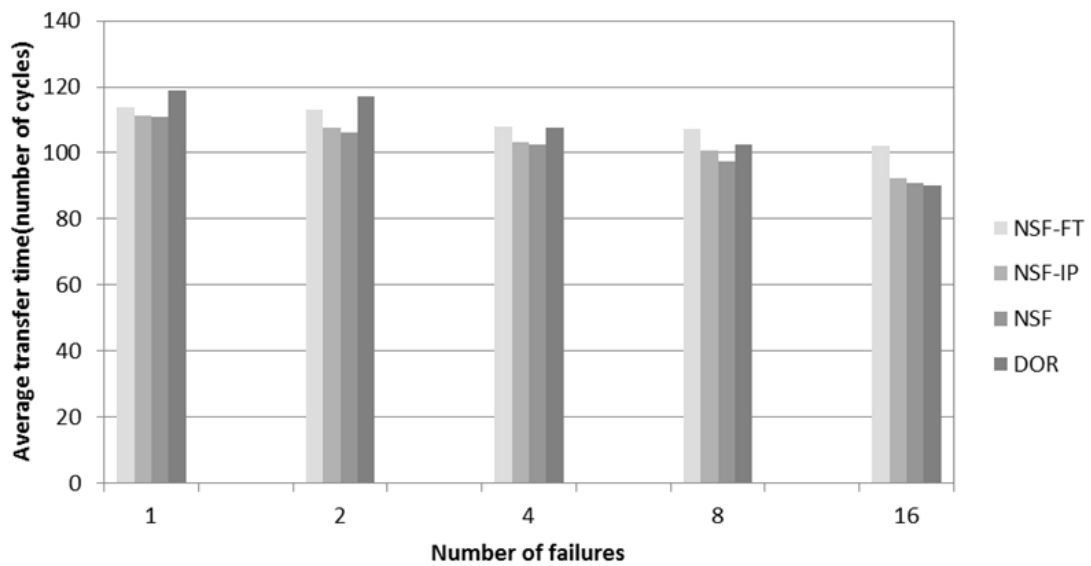


図 27 ループ数 5 における平均遅延時間

表 13 NSF-FT と他手法の不着数を比較した割合

	Number of failures	Loop1	Loop3	Loop5
DOR→FT	1	69.2%	54.0%	83.2%
	2	70.5%	70.1%	82.2%
	4	67.0%	75.1%	88.6%
	8	72.1%	82.3%	92.2%
	16	76.1%	92.0%	94.3%
IP→FT	1	60.0%	87.2%	94.0%
	2	99.3%	85.9%	93.2%
	4	77.6%	90.8%	97.7%
	8	101.8%	91.7%	98.0%
	16	93.8%	100.1%	97.3%
NSF→FT	1	64.3%	57.4%	89.9%
	2	91.4%	82.5%	89.8%
	4	94.5%	88.4%	94.9%
	8	91.1%	88.9%	96.5%
	16	89.9%	98.8%	96.3%

第5章 まとめ

本論文では、NSF 法を改善し、耐故障性を改善した新たなルーティング・アルゴリズムとして、NSF-IP 法、および NSF-FT 法の提案、評価を行った。NSF-IP は、NSF 法における最短経路の制約を排除することにより、より自由度を高め、耐故障性を向上させたアルゴリズムである。NSF-FT は、NSF-IP に加えて、仮想チャネルの移動を許すことによりさらに経路選択の自由度を高めたものである。本論文においては、これらの手法の詳細を説明するとともに、シミュレーションによる動的通信性能評価により、NSF-IP、NSF-FT それぞれについて、耐混雑性と耐故障性の評価を行った。

NSF-IP に関して、性能評価として、耐混雑性能の評価を Uniform Traffic Pattern と Matrix Transpose により行い、提案アルゴリズムが DOR と比較して優れていることを明らかにした。この結果を踏まえた上で耐故障性能に関する評価も行った。1 カ所の故障、4 隅の故障、中央 4 カ所の故障、1 箇所から 2、4、8、16 箇所のランダムな故障による評価を行った結果、NSF-IP 法が NSF 法、DOR に比べ耐故障性能が優れていることが分かった。本稿の手法により、NSF-IP 法は一定程度の耐故障性を有することが明らかになった。

NSF-FT の耐混雑性については、DOR に比べスループットは改善しており、NSF、NSF-IP

と比較した場合も実装によるスループットの低下はみられなかった. 故障 PE を想定した耐故障性の評価については DOR, NSF 法, NSF-IP 法, 本稿で提案した NSF-FT の順にパケットの到達率が上昇し, 耐故障性に優れていることが明らかになった.

今後, より高い耐故障性を有するルーティング・アルゴリズムの実現のため, 故障 PE を矩形領域でまとめて故障領域として扱う手法を組み合わせることにより, さらなる耐故障性の向上を実現する予定である.

謝辞

本研究の方針など, 多大なご指導をいただきました三浦康之教授に感謝いたします. また, 論文やプレゼンテーションの資料などにおいて様々なご助言をいただきました渡辺重佳教授, ご多忙の中審査に立ち会っていただいた二宮洋教授, 中上川友樹教授に心よりお礼申し上げます. 研究生活の日々を共に過ごしてくださった三浦研究室の皆様に感謝致します. また, 私の相談を聞いてくださった友人達に感謝致します. 最後に, ここまでの研究生生活を支えてくださった父と母, そして二人の姉に感謝致します.

参考文献

- [1] J.Y. Ngai, C.L. Seitz : “A framework for adaptive routing in multicomputer networks”, ACM SIGARCH Computer Architecture News, Vol.19, No.1, pp.6-14 (1991)
- [2] T. Schonwald, J. Zimmermann, O. Bringmann, W. Rosenstiel : “Fully Adaptive Fault-Tolerant Routing Algorithm for Network-on-Chip Architectures”, Digital System Design Architectures, Methods and Tools, pp.527-534 (2007)
- [3] M.M. Hafizur Rahman, Yukinori Sato, Yasuyuki Miura, and Yasushi Inoguchi : “Dynamic Communication Performance of a Hierarchical 3D-Torus Network”, IASTED, In 10th International Conference Parallel and Distributed Computing and Networks (PDCN 2011) (2011)
- [4] Y. Miura and S. Horiguchi : “An Adaptive Routing for Hierarchical Interconnection Network TESH”, Proc. of the Third International Conference on Parallel And Distributed Computing, Applications and Technologies, pp. 335-342 (2002)
- [5] Y. Miura, Masahiro Kaneko, M.M.Hafizur Rahman and Shigeyoshi Watanabe : “ Adaptive Routing Algorithms and Implementation for TESH Network”, Communications and Network (CN), Vol.5, No.1, pp.34-49 (2013)
- [6] W.J. Dally : “Performance Analysis of k-ary n-cube Interconnection Networks, IEEE Trans. on Computers”, vol. 39, No.6, pp.775-785 (1990)
- [7] W.J. Dally and Hiromichi Aoki : “Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels”, IEEE Trans. On Parallel and Distributed Systems, Vol.4, pp. 466-475 (1993)
- [8] M.P.Merlin and J.P.Schweitzer : “Deadlock Avoidance in Store-and-Forward Networks-1: Store and Forward Deadlock”, IEEE Trans. On Communications”, Vol.COM-28, No.3, pp.345-354 (1980)
- [9] W.J.Dally and C.L.Seitz : “Deadlock-Free Message Rouring in Multiprocessor interconnection Networks. ”, IEEE Trans. On Computers, Vol. C-36, No.5, pp.547-553 (1987)
- [10] C.S.Yang and Y. M. Tsai : “Adaptive Routing in k-ary n-cube Multicomputers”, Proc. of 1996 International Conference on Parallel and Distributed Systems(ICPADS'96), pp. 404-411 (1996)
- [11] J.Duato : “A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks”, IEEE Trans. on Parallel and Distributed Systems, Vol.4, No.12, pp.1320-1331 (1993)
- [12] D.H. Linder and J.C. Harden : “An adaptive and fault tolerant wormhole routing

- strategy for k-ary n-cubes”, IEEE Trans. on Computers, vol.C-40, No.1, pp.2-12 (1991)
- [13] R.S. Ramanujam, Bill Lin : “Destination-based adaptive routing on 2D mesh networks”, 2010 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), pp.1-12 (2010)
- [14] C.J.Glass and L. M. Ni : “Maximally Fully Adaptive Routing in 2D Meshes”, Proc. of The 19th International Symposium on Computer Architecture, pp. 278-287 (1992)
- [15] C.J.Glass, L.M.Ni : “The Turn Model for Adaptive Routing”, Proc. of The 25th Annual International Symposium on Computer Architecture, pp.441-450 (1998)
- [16] Jie Wu : “A Fault-tolerant and Deadlock-free Routing Protocol in 2D Meshes Based on Odd-even Turn Model”, IEEE Trans. on Computers, Vol.52, No.9, pp.1154-1169 (2003)
- [17] A.Jouraku, M.Koibuchi, H.Amano : “An Effective Design of Deadlock-Free Routing Algorithms Based on 2D Turn Model for Irregular Networks”, IEEE Trans. on Parallel and Distributed Systems, Vol.18, No.3, pp.320-333 (2007)
- [18] W.J. Dally : “Virtual-Channel Flow Control”, IEEE Trans. on Parallel and Distributed Systems, Vol.3, No.2, pp.194-205 (1992)
- [19] Y.Miura, K.Shimozono, K.Matoyama, and S.Watanabe : “An Adaptive Routing of the 2-D Torus Network Based on Turn Model”, Proc. of 4th International Workshop on Advances in Networking and Computing, pp.587-591 (2013)
- [20] Yasuyuki Miura, Kentaro Shimozono, Kazuya Matoyama, and Shigeyoshi Watanabe : “The Static and Dynamic Performance of an Adaptive Routing Algorithm of 2-D Torus Network Based on Turn Model”, Proc. of the 2014 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA' 14), pp.114-120 (2014)
- [21] Yasuyuki Miura, Kentaro Shimozono, Naohisa Fukase, Shigeyoshi Watanabe, and Kazuya Matoyama : “An Adaptive Routing Algorithm of 2-D Torus Network Based on Turn Model: The Communication Performance”, International Journal of Networking and Computing (IJNC), pp.223-238 (2015)
- [22] J.Duato : “A necessary and Sufficient Condition for Deadlock-Free Adaptive Routing Wormhole Networks”, Proc. of the International Conference on Parallel Processing, Vol.1, pp.142-149 (1994)
- [23] E. Fleury and P.Fraigniaud : “A General Theory for Deadlock Avoidance in Wormhole-Routing Networks”, IEEE Trans. on Parallel and Distributed Systems, Vol.9, No.7, pp.626-638 (1998)

- [24] N.Aosaka, Y.Fukushima, M.Fukushi, and M.Kameyama : “Fault-Tolerant Cogestion-Avoidance Routing for 2D-Mesh Network-on-Chip”, Technical report of IEICE. FIIS10, no. 271, March (2010)
- [25] Naohisa Fukase, Yasuyuki Miura, Shigeyoshi Wwatanabe, and M.M. Hafizur Rahman : “The Performance Evaluation of a 3D Torus Network Using Partial Link-Sharing Method in NoC Router Buffer”, IEICE Trans. on Information and Systems, Vol.E100–D,no.10,October (2017)
- [26] Naohisa Fukase, Yasuyuki Miura, Shigeyoshi Wwatanabe : “Link-Sharing Method of Buffer in Router Circuit of Direct-Connection Network”, IEEJ Trans EIS, vol 132,No.10, (2012)(in Japanese)
- [27] Tsukasa-Pierre Nakao, Yasuyuki Miura : “The Study on Adaptive Routing Algorithm of 2-D Torus Network with Fault Tolerance”, IEICE Technical Report (FIIS64),October (2017).

研究業績

・論文

- 1) 中尾司ピエール, 三浦康之, 深瀬尚久, Turn モデルに基づく二次元トーラス・ネットワークの適応ルーティング・アルゴリズムにおける耐故障性評価, 電気学会論文誌 C 分冊, (採録決定)
- 2) Tsukasa-Pierre Nakao, Yasuyuki Miura, and Naohisa Fukase, Fault-tolerant adaptive routing algorithm for 2D torus network, Transactions on Networks and Communications (採録決定)

・国際会議

- 1) Tsukasa-Pierre Nakao, Yasuyuki Miura, and Naohisa Fukase, The Study on Adaptive Routing Algorithm of 2-D Torus Network with Fault Tolerance, IEEE International Conference on Consumer Electronics – Taiwan (IEEE 2018 ICCE-TW), pp.105-106, 2018.05.

・口頭発表

- 1) 中尾司ピエール, 三浦康之, 耐故障性能を考慮した二次元トーラス・ネットワークの適応ルーティングアルゴリズムの検討, 電子情報通信学会技術研究報告, FIIS-17-64, pp.1-10, 2017.10.

・その他の論文等

- 1) 中尾 司ピエール, 坂下 善彦, 「SNMP によるネットワークシステム管理のための報告型情報収集方式」, 情報処理学会第 73 回全国大会講演論文集, pp. 427-428, 2011 年 3 月
- 2) 中尾 司ピエール, 坂下 善彦, 「Linux 仮想化環境におけるメモリコミットの分析」, 研究報告マルチメディア通信と分散処理 (DPS) 2012-DPS-152(32), pp. 1-5, 2012 年 9 月
- 3) 中尾 司ピエール, 坂下 善彦, 「Linux 仮想化環境における VM システムメモリ使用量推測と評価」, 情報処理学会第 75 回全国大会講演論文集, pp. 213-214, 2013 年 3 月
- 4) 中尾 司ピエール, 坂下 善彦, 「仮想環境下におけるゲスト OS 内共有ライブラリのホスト OS による共通化」, 情報処理学会第 76 回全国大会講演論文集, pp. 165-166, 2014 年 3 月
- 5) 中尾 司ピエール, 坂下 善彦, 大谷 真, 「仮想計算機間におけるライブラリ重複に注目したホスト計算機資源の縮小化の検討」, 情報処理学会第77回全国大会講演論文集, pp. 13-14, 2015年3月

・その他国際会議

- 1) Tsukasa-Pierre Nakao, 「Provisional process migration from IoT devices: A performance study」
, 2015 IEEE 4th Global Conference on Consumer Electronics,,pp.490-491,2018.10.