特別研究報告

題目

データセンターにおける負荷分散を実現する
ネットワーク構成手法および経路制御手法の評価

指導教員
村田 正幸 教授

報告者
下間 雄太

平成 24 年 2 月 16 日

大阪大学 基礎工学部 情報科学科

平成 23 年度 特別研究報告

データセンターにおける負荷分散を実現する
ネットワーク構成手法および経路制御手法の評価

下間 雄太

内容梗概

ネットワークを介したサービスが多様化し，データセンター内で処理されるデータ量が増加している．データ量の増加に伴い，それを処理するデータセンター内のサーバー数も増加しており，1 箇所のデータセンターで，数万台から数十万台のサーバーが収容されるようになってきている．データセンターでは，サーバー間の通信性能がデータセンターの処理性能に大きな影響を与えるため，データセンターに適したネットワークを構築・制御する手法に注目が集まっている．中でも，データセンター内で頻繁に発生する機器の故障，短い間隔で発生する予測不可能なトラヒック変動といった環境変動への対応は大きな課題となっている．

環境変動に対応する方法として，環境変動に合わせてネットワーク内の経路を変更する手法が考えられる．しかしながら，データセンター内では，環境変動が激しく，従来通信ネットワークで研究が進められてきたトラヒックエンジニアリングのようなネットワーク全体のトラヒックを把握して経路を計算するような制御では，環境変動に対応することが困難である．そのため，大規模なデータセンターでは，局所的な情報のみを用いて経路制御を行うことが必要であると考えられる．しかしながら，データセンターネットワーク内の経路制御に関する既存研究は，特定のネットワーク構成を対象としており，どのようなネットワーク構成が局所的な情報を用いた経路制御に適しているのかは明らかにされていない．

本報告では，ポート数が等しいスイッチを同数用いて構築可能なネットワーク構成を複数生成し，局所的な情報を用いた経路制御手法や，ランダムに転送先を選択することにより負荷分散を行う経路制御手法，ネットワーク全体のトラヒック情報を用いた経路制御手法を動作させ，収容可能なトラヒック量，サーバ間に確保可能な帯域の比較を行うことにより，データセンターに適したネットワーク構成と経路制御手法の組み合わせを明らかにする．評価の結果，ネットワーク全体のトラヒック情報を用いた経路制御手法は，トラヒック変動発生後はランダムな経路選択を行う手法と同程度のトラヒックしか収容できないことが明らかとなり，頻繁なトラヒック変動に対応するためには，局所的な経路制御が必須であることが確認できた．また，局所的な情報を用いた経路制御を動作させた場合，下位層のネットワー

ク構成を複数相互接続することにより上位層のネットワーク構成を構築するという階層的な手順でネットワークを構成し，かつ，各スイッチのポートのうち各階層の接続に用いられるポート数が均等となる構成が，構築されたネットワークの規模によらず，最も多くのサーバ間トラヒックを収容でき，故障発生時にも収容できるトラヒック量を維持できることが明らかになった．

主な用語

データセンター，負荷分散，ネットワーク構成手法，経路制御手法

目 次

図 目 次

表 目 次

1　はじめに

近年，クラウドコンピューティングなど，データセンターを介して提供されるサービスが増加し，データセンター内で処理されるデータも著しく増加している．処理が必要なデータが増加するにつれ，データセンター内のサーバー数も増加し，数万台から数十万台のサーバーを収容する大規模なデータセンターも構築されるようになってきた．

データセンターでは，多くのサーバーが連携することにより多量のデータの処理を行っている．たとえば，Google などの検索エンジンにおいては，MapReduce [1] と呼ばれる手法を用い，多数のサーバーの連携により，WEB の検索データベースの更新が行われている．また，Google File System [2] では，多数のサーバーが連携することにより，膨大なデータを保持している．

データセンター内の処理は複数のサーバーの連携により行われるため，サーバー間を接続するネットワークの性能がデータセンターの処理性能に大きな影響を与える．しかしながら，従来のデータセンター内で用いられていた木構造のネットワークは，サーバー間に十分な通信帯域を供給できないなどの問題があり，大規模なデータセンターへの適用は困難である．そのため，大規模なデータセンターに適したネットワーク構成が必要とされている．

また，データセンターでは，機器の故障や予測不可能なトラヒック変動 [3–5] が頻繁に発生する．データセンターの処理性能を維持するためには，故障やトラヒック変動といった環境変動が発生した際にも，ネットワーク内の一部に負荷が集中することは避け，サーバー間に十分な通信帯域を維持することが必要となる．そのためには，故障やトラヒック変動が発生してもサーバ間の接続性が維持可能なネットワーク構成を構築するのみならず，環境変動に合わせてネットワーク資源を効率的に利用できるように経路を設定することが必要となる．しかしながら，短い間隔で発生する環境変動に対応して，ネットワーク全体の情報を収集し，集中的に経路制御を行うことは困難である．そのため，データセンター内の経路制御は，ネットワーク内の各地点で把握可能な局所的な情報を元に行うことが望ましいと考えられる．また，ネットワーク構成によっては，ネットワーク全体の情報を得ずに経路制御をすると特定のリンクに負荷が集中する可能性もあり，局所的な情報を用いた経路制御の有効性は，適用先のネットワーク構成にも依存すると考えられる ．そのため，データセンターのネットワーク構成も，局所的な情報のみを用いて環境変動への対応を行う経路制御手法が有効に動作可能な構造が望ましいと考えられる．

近年，データセンターに適したネットワークの構成手法に関する様々な研究が進められている [6–12]．文献 [6] では，ポート数の少ない安価なスイッチのみを用いて，十分な帯域を全サーバー間に確保可能な FatTree と呼ばれるネットワークを構成する手法が提案されており，文献 [9] では，ポート数の多いスイッチを用いることにより，全サーバー間に十分な帯

域を確保しつつ，サーバー間の平均ホップ数が小さくなるようなネットワーク構成手法が提案されている．また，文献 [7, 8] では，複数の NIC を持つサーバーを用い，サーバーとスイッチだけでなく，サーバー間も直接接続することにより，少ない機器数・低消費電力で大規模なデータセンターネットワークを構築する手法が提案されている．さらに我々の研究グループにおいても，データセンター内のアプリケーションの要求に合わせてパラメーターを設定することにより，要求を満たす様々なネットワークを構築可能な Generalized Flattened Butterfly (GFB) [13] と呼ばれる構造を提案している．

また，データセンターネットワーク上で経路を制御する手法についても検討が進められている [5, 10, 14]．Tree 型の構造データセンターネットワークを対象として，ランダムに転送先の上位スイッチを選択することによる負荷分散を行う手法 [14]，サーバー間に本来流れるトラヒック量を推定した上で，そのトラヒックを収容可能な経路を集中制御により計算する手法 [10] が提案されている．また，文献 [5] では，データセンターネットワーク内において全地点間のトラヒック量を把握し，そのトラヒック量に合わせて経路を最適化する手法の有効性が議論されており，1 秒間隔で最適制御を行うことができればトラヒック変動に対応できることが明らかにされている．

しかしながら，これらの研究では，データセンターネットワークの構成手法に関する研究や，データセンター内の経路制御に関する研究が独立した研究として行われており，データセンターネットワークの構成が経路制御に与える影響や，データセンター内で起こりうる環境変動に対応した経路制御を行うのに適したネットワーク構成は明らかになっていない．

本報告では，データセンターネットワークの構成と経路制御手法の組み合わせを評価し，トラヒック変動や故障などが発生した場合にも性能を維持可能な適切な組み合わせを明らかにする．本評価においては，FatTree と合わせて，GFB のパラメーターを変化させることにより生成した様々なネットワーク構成を評価対象のネットワーク構成として用いる．そして，各ネットワーク構成上で，集中制御により経路を決めた場合，トラヒック情報を用いずにランダムな負荷分散を行った場合，局所的なトラヒック情報を用いて経路を決めた場合に，サーバー間に確保可能な通信帯域やネットワーク内に収容可能なサーバー間トラヒック量の比較を行う．また，各ネットワーク構成・経路制御手法の組み合わせに対して，トラヒック変動が発生した場合や故障が発生した場合にネットワーク内に収容可能なトラヒック量についても評価を行い，環境変動発生時にもサーバー間に十分な通信帯域を確保可能なネットワーク構成と経路制御手法の組み合わせを明らかにする．

本報告の構成は以下の通りである．まず 2 章では，データセンターにおけるネットワーク構成手法について述べる．次に 3 章では，データセンターネットワークにおける経路制御手法を，トラヒック情報を用いない経路制御手法，局所的なトラヒック情報を用いる経路制御手法，全体のトラヒック情報を用いる経路制御手法に分けて議論する．4 章では本報告

で行う評価の概要と評価指標を説明し，5 章で評価結果を述べる．最後に，6 章において，本報告のまとめと今後の課題について述べる．

2　データセンターにおけるネットワーク構成手法

本章では，従来研究で提案されているネットワーク構成手法，および，我々の研究グループで提案しているパラメーター設定により様々なデータセンターネットワークを生成可能なGFB と呼ばれるデータセンターのネットワーク構成手法について説明する．

2.1　データセンターにおける従来型ネットワーク構成手法

2.1.1　FatTree

文献 [6] では，ポート数が少ない安価なスイッチのみを用いて，全サーバー間に十分な帯域を確保可能な FatTree と呼ばれるネットワークを構成する手法が提案されている．FatTree は，図 1 のように，ポート数が少ないスイッチを組み合わせて POD と呼ばれる構造を構築し，POD の各ポートを最上位層に配置されたスイッチに接続することにより構築される．

各 POD は，ポート数の少ないスイッチを用いて仮想的に構築されたポート数の多いスイッチである．各 POD は Butterfly と呼ばれるネットワーク構造を用いて構築される．Butterfly では，スイッチが階層的に配置され，各階層のスイッチは，そのポートの半分を上位層のスイッチとの接続，残り半分を下位層のスイッチとの接続に用いる形で，最下位層に配置された各スイッチから最上位層に配置された全スイッチに到達可能なように接続される．そして，最下位層に配置されたスイッチにサーバーを接続する．

文献 [6] では，スイッチを 2 階層に配置して POD を構成した，全 3 階層の構造を提案していた．しかしながら，スイッチを 3 階層以上に配置した POD を構成することも可能であり，階層数を増やすことにより，より多くのサーバーを収容できるネットワークを構築できる．

多階層型の FatTree において，最下位層のスイッチの台数が n 台となる構成をポート数 p のスイッチを用いて作るために必要な階層数は以下のように求めることができる．

$$k = \begin{cases} \lceil \log_{\frac{p}{2}} \left(\frac{n}{p} \right) + 2 \rceil (p < n) \\ 2 (n \leq p) \end{cases} \tag{1}$$

その後，最上位層以外の層に n 個のスイッチを配置，最上位層に $\lfloor \frac{n}{2} \rfloor$ 個のスイッチを配置し，各階層間のリンクを構築することにより，最下位層を n 台のスイッチで構成した FatTree を構築することができる．

FatTree は各サーバー間に $\frac{p}{2}$ 本の独立した経路があるため，サーバー間に大きな帯域を確保することができ，故障時にもサーバー間の接続性を維持できる．しかしながら，多くのサーバーを接続するためには，階層数を増やすか，ポート数の多いスイッチが必要となり，消費電力が大きくなってしまうという問題がある．

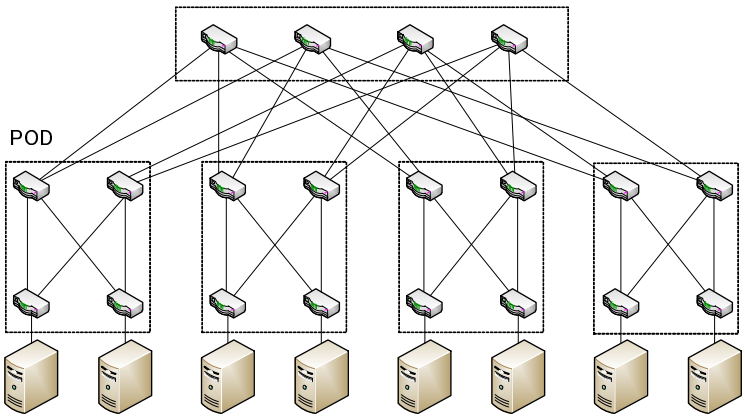


図 1: FatTree

2.1.2 DCell

文献 [7] では，複数の NIC を持つサーバー間を接続することにより，大規模なデータセンターネットワークを構築可能な DCell と呼ばれるネットワーク構成手法が提案されている．DCell は階層的に構築される構造であり，上位層の DCell は複数の下位層の DCell を相互接続することで構築される．以降，k 層目の DCell を DCell_k と表記する．また，k 層目の DCell に含まれるサーバーの台数を N_k とする．

図 2 に DCell の例を示す．DCell_0 は n 台のサーバーを 1 台のスイッチを介して接続することで構成される．DCell_k は $\prod_{i=0}^{k-1} N_i + 1$ 個の異なる DCell_{k-1} を相互接続することにより構築される．DCell_{k-1} の相互接続は，サーバーのポート同士を直接接続することにより構成される．DCell_k を構築する際には，各 DCell_{k-1} と各 DCell 内のサーバーに ID を割り振り，その ID を基準に DCell_{k-1} 間の接続に用いられるサーバーを選択する．ID が i の DCell_{k-1} 内のサーバーのうち，ID が j の DCell_{k-1} との接続には，$i < j$ の場合はサーバーの ID が $j-1$ のサーバー，$j > i$ の場合はサーバーの ID が j のサーバーが用いられる．

DCell では，サーバーに複数のポートを設けることで少ないスイッチ数で大規模なネットワークを構築することができる．しかしながら，サーバーの NIC の帯域が他のサーバー間の通信にも利用されるため，各サーバーの NIC の帯域のすべてを当該サーバーの通信に利用することができず，サーバー間の通信に十分な帯域が確保できない可能性もある．

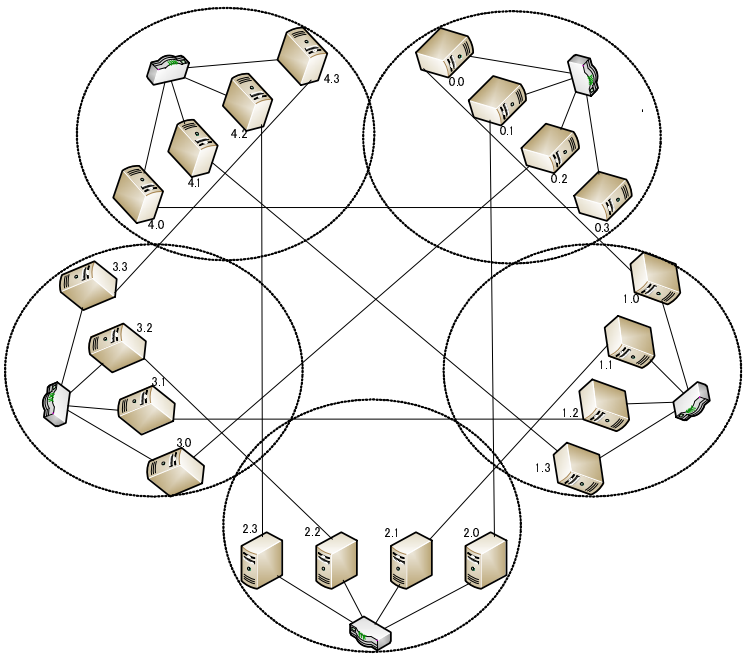


図 2: DCell

2.1.3 Flattened Butterfly

文献 [9] では，Flattened Butterfly と呼ばれる，ポート数の多いスイッチを用いて，全サーバー間に十分な帯域を確保しつつ，サーバー間の平均ホップ数を小さくすることができるネットワークを構成する手法が提案されている．Flattened Butterfly は，FatTree の POD を構成する際にも用いられていた Butterfly を元にしたもので，図 3 のように Butterfly の異なる階層のスイッチを一つにまとめることにより構築される．

Flattened Butterfly も以下のように階層的に構築された構造とみなすことができる．最下位層の Flattened Butterfly は，その階層の Flattened Butterfly を構築するのに用いられる全スイッチ間を接続することにより構築される．そして，k 層の Flattened Butterfly は各スイッチから，他の全ての $k-1$ 層 Flattened Butterfly にリンクを構築することにより，$k-1$ 層 Flattened Butterfly を相互に接続することにより構築される．

Flattened Butterfly では，スイッチのポート数と同数の独立した経路が各スイッチ間に存在するため，各スイッチ間に十分な帯域を確保することができる．しかしながら，Flattened Butterfly で規模の大きなネットワークを構築する際には，ポート数の多いスイッチが必要となる．

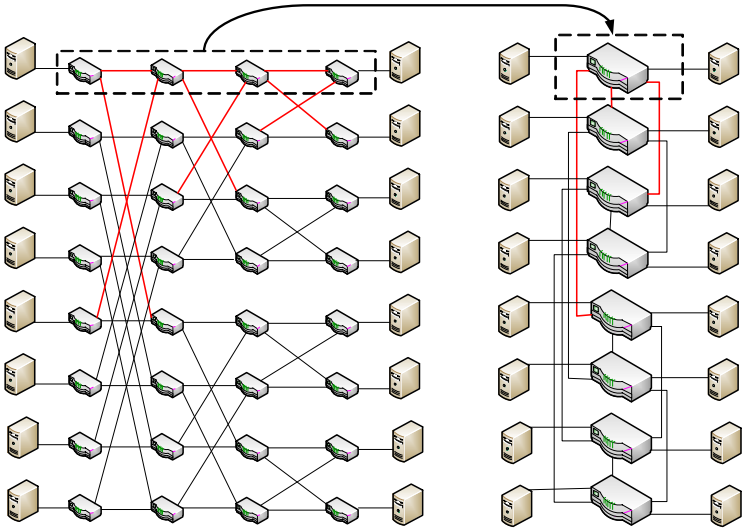


図 3: Flattened Butterfly

2.1.4 Torus

Torus はスーパーコンピューター内で用いられているネットワーク構成であり [15,16]，図 4 のように多次元の格子状にスイッチを並べ，隣接したスイッチを接続，格子の両端のスイッチを接続したネットワーク構成となる．

k 次元の Torus は，$2k$ ポートのスイッチを用いて構築可能である．また，低次元の Torus においても，接続可能なスイッチ数に制限はないため，ポート数の少ないスイッチのみを用いた場合でも大規模なネットワークを構築することが可能となる．しかしながら，次数 k，各次元におけるスイッチ数が n_i である Torus 内のスイッチ間最大ホップ数は $\sum_{i=1}^{k} \lfloor \frac{n_i}{2} \rfloor$ であり，収容スイッチ数が大きくなると最大ホップ数も大きくなる．

2.2 Generalized Flattened Butterfly

従来のデータセンターネットワーク構成は，大きな帯域を確保できる構成ではポート数の多いスイッチが必要になる，あるいは，必要なスイッチ数が多くなるため，消費電力が大きくなる．また，消費電力が少ないネットワーク構成はサーバー間に十分な帯域を確保できない可能性がある．そのため，データセンターネットワークの消費電力を抑えるためには，データセンターネットワーク内を流れるトラヒック量やアプリケーションの要求に合わせて，適切なデータセンターネットワークの構成を選択する必要がある．

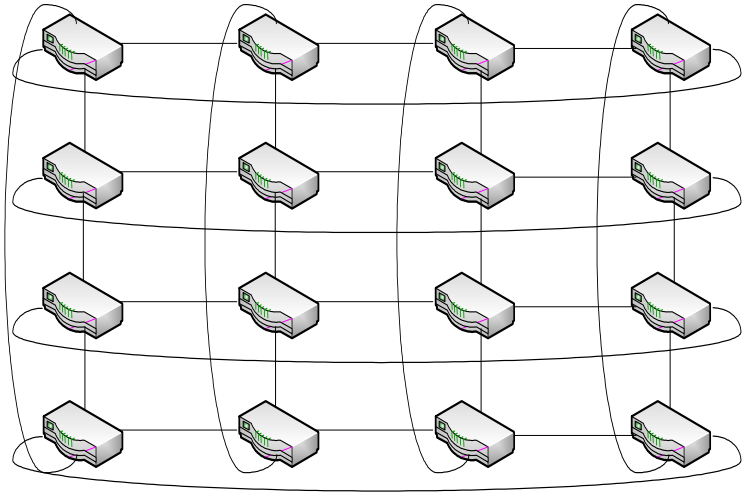


図 4: 2 次元 Torus

我々の研究グループでは，データセンターネットワークの要求に合わせて，ネットワーク構成を設定できる構造として，Generalized Flattened Butterfly (GFB) [13] と呼ばれる構造を提案している．Generalized Flattened Butterfly は，そのパラメーターの設定により，スイッチ間の接続に必要なリンク数を設定することができる構造で，Torus，Flattened Butterfly や，最下位層をフルメッシュに接続したスイッチに置き換えた DCell を含む様々なネットワーク構成を構築することができる．そして，文献 [13] では，GFB のパラメーターを動的に設定することにより，データセンター内のトラヒックの変化や要求の変化に対応して，適切なネットワーク構成に移行する手法が提案されている．

GFB は階層的に構築されるネットワーク構成である．k 層で構成される GFB を GFB_k と置くと，GFB_k は複数の GFB_{k-1} を相互接続することにより構築される．GFB では，以下のパラメーターによりネットワーク構成を設定できる．

- 階層数：k
- 階層 k で利用する各スイッチあたりのリンク数：L_k
- 階層 k の GFB に所属する GFB_{k-1} の数 (階層 1 においてはスイッチ数) ：N_k
- 階層 k で同一の GFB_{k-1} 間の接続に用いるリンクの最小本数：M_k

たとえば，$L_k = N_k - 1$ となるように定めた GFB は，Flattened Butterfly 構造となり，$L_1 = N_1 - 1$，$L_k = 1 (k > 1)$ と定めた GFB は 最下位層をフルメッシュに接続したスイッ

チに置き換えた DCell となる． また，$L_k = 2$，$M_k = \prod_{i=1}^{k-1} N_i$ と定めた GFB は Torus となる．

GFB_k は以下の 2 つの手順により GFB_{k-1} を相互接続することにより構築することができる．

- 各 GFB_{k-1} 間のリンクの本数の決定
- 各 GFB_{k-1} 間のリンクを追加するスイッチペアの決定

上記の手順を行うにあたり，各階層の GFB に一意な ID を割り当て，GFB の ID を元に GFB_{k-1} 間の接続を決定する．各スイッチは所属する各階層の GFB の ID の組み合わせで識別できるものとする．以降，スイッチ s が所属する k 層の GFB の ID を $D_k^{gfb}(s)$ とする．また，k 層の GFB 内でのスイッチ s の ID は GFB の ID を元に，以下のように計算した値 $D_k^{sw}(s)$ で与えるものとし，このスイッチの ID を元に GFB_{k-1} 間の接続に用いられるスイッチを決定する．

$$D_k^{sw}(s) = \sum_{1 \leq i < k} D_i^{gfb}(s) n_{i-1}^{gfb} \tag{2}$$

ただし，n_k^{gfb} は，k 層の GFB に所属するスイッチの数であり，以下の式で求めることができる．

$$n_k^{gfb} = \begin{cases} \prod_{i=1}^{k-1} N_i (0 < k) \\ 1 (k = 0) \end{cases} \tag{3}$$

各 $k-1$ 層の GFB 間のリンクの本数の確定　各 GFB_{k-1} 間の接続構成は次の手順で構築される．

手順 1　各 GFB_{k-1} が他の GFB_{k-1} との接続に用いることができるリンク数 L_k^{gfb} を求める．L_k^{gfb} は以下の式で求めることができる．

$$L_k^{gfb} = L_k \prod_{i=1}^{k-1} N_i \tag{4}$$

手順 2　各 GFB_{k-1} が他の GFB_{k-1} との接続に用いることができるリンク数を M_k で割った値 $\frac{L_k^{gfb}}{M_k}$ が $(N_k - 1)$ 以上であれば，全 GFB_{k-1} との間に $\lfloor \frac{L_k \prod_{i=1}^{k-1} N_i}{M_k (N_k - 1)} \rfloor M_k$ 本のリンクを構築する．

手順 3　各 GFB_{k-1} が他の GFB_{k-1} との接続に用いることができるリンクの本数を M_k で割った値 $\frac{L_k^{gfb}}{M_k}$ が $(N_k - 1)$ 未満であれば，GFB の ID が隣り合う GFB_{k-1} を M_k 本のリンクで接続する．

手順 4　各 GFB_{k-1} が他の GFB_{k-1} との接続に用いることができるリンクの本数 L_k^{gfb} から，手順 1・手順 2 で追加されたリンクの本数を除いた値 L'_k を計算する．L'_k が正であれば，$D_k^{gfb}(s)$ という ID を持つ GFB_{k-1} と $D_k^{gfb}(s)+Cp_k$ $(C=1,2,\cdots,L'_k)$ に該当する ID を持つ GFB_{k-1} を接続する．

手順 4 において，p_k は以下のように定めることで，各 GFB_{k-1} の接続先となる GFB_{k-1} の ID が等間隔となるようにしている．

$$p_k = \left\lfloor \frac{N_k}{L'_k+1} \right\rfloor \tag{5}$$

これにより，GFB_{k-1} 間の最大ホップ数が小さくなるようなリンクを追加することができる．

各リンクに接続するスイッチの選択　各 GFB_{k-1} 間の接続リンクの本数が確定した後，以下の手順により，具体的にどのスイッチ間にリンクを構築するのかを決定する．GFB の ID が $D_k^{gfb}(a)$ の GFB_{k-1} と $D_k^{gfb}(b)$ の GFB_{k-1} の間を結ぶリンクが接続されるスイッチは以下の手順で選択される．

手順 1　$D_k^{gfb}(b)$ が，$D_k^{gfb}(a)$ の接続する GFB_{k-1} の中で何番目に GFB の ID が小さいかを求める．求めた値を $R(D_k^{gfb}(b))$ とする．

手順 2　以下の式を満たすスイッチ s を $D_k^{gfb}(b)$ の ID を持つ GFB_{k-1} との接続に用いるスイッチの候補とする．

$$D_k^{sw}(s) = R(D_{k-1}^{gfb}(b)) + \left\lfloor \frac{Cn_{k-1}^{gfb}}{l_{(D_{k-1}^{gfb}(a),D_{k-1}^{gfb}(b))}} \right\rfloor (C=0,1,\cdots,l_{(D_{k-1}^{gfb}(a),D_{k-1}^{gfb}(b))}-1)$$

ただし，n_{k-1}^{gfb} は，GFB_{k-1} に所属するスイッチの数である．$l_{(D_{k-1}^{gfb}(a),D_{k-1}^{gfb}(b))}$ は ID が $D_{k-1}^{gfb}(a)$ と $D_{k-1}^{gfb}(b)$ の GFB_{k-1} 間で接続に用いられるリンクの本数である．

手順 3　接続スイッチの候補に含まれているスイッチで，$D_{k-1}^{gfb}(b)$ の ID を持つ GFB_{k-1} との接続に用いられるリンク数が最も少ないスイッチを選択する．

手順 4　手順 3 で選択したスイッチに接続している GFB_{k-1} 間のリンク数が L_k よりも少ないなら，そのスイッチを接続先として確定する．そうでなければ，GFB_{k-1} 間のリンク数が L_k よりも少ないスイッチのうち手順 3 で選択したスイッチの ID に最も近い ID を持つスイッチを選択する．

3　データセンターネットワークにおける経路制御手法

本章では，データセンターネットワークにおける経路制御手法を，トラヒック情報を用いない経路制御手法，局所的なトラヒック情報を用いる経路制御手法，全体のトラヒック情報を用いる経路制御手法の 3 つの手法に分けて説明する．

3.1　トラヒック情報を用いない経路制御手法

データセンターネットワークにおける経路制御手法の一つに，トラヒック情報を用いず，ネットワークの構造から経路を確定する手法がある．この方法としては，最短ホップの経路を用いる方法や，DCell の構造を利用し，DCell 内の各サーバーに割り振られた ID を元に経路を決める DCell Routing [7] などがある．しかしながら，この手法は，負荷分散を行わないため，一部のサーバーラック間にのみ大きなトラヒックが流れるなどのトラヒックの偏りがある場合，一部のリンクに負荷が偏ってしまい，サーバー間に十分な帯域を確保することができない可能性がある．

トラヒック情報を用いずに負荷分散を行う手法も存在する [17,18]．これらの手法では，候補となる経路にランダムにトラヒックを収容することによって負荷分散を行う．以降，本報告では，ランダムにトラヒックを収容することによって負荷分散を行う手法をランダム制御と呼ぶ．

ランダム制御では，各スイッチにおいて転送先候補の中からランダムに転送先を選択することで，一部のリンクに負荷が集中することを防ぐことができる．また，ランダム制御では，トラヒック情報を収集することなく制御を行うため，トラヒック変動発生時にも，トラヒック情報の収集や適切な経路の計算にかかる時間を待つことなく，継続して負荷分散を行うことができる．しかし，現在流れているトラヒック量を考慮入れた制御ではないため，トラヒック情報を用いた経路制御と比べると，効率的な負荷分散を行うことはできず，サーバー間に確保できる帯域も小さくなる．

3.2　局所的なトラヒック情報を用いる経路制御手法

データセンターの経路制御手法には，各スイッチで把握可能なトラヒック情報を用いて，各スイッチが自律的に転送先を判断する手法も存在する．この手法では，各スイッチが，接続しているリンクを流れるトラヒック量やキュー長をリアルタイムで把握できる状況を考え，把握した情報を元に，転送先の候補のうち，最も負荷が低い転送先や，最も転送先への遅延が少なくなると予測される転送先を各スイッチが自律的に選択する [19]．以降，この手法を局所制御と呼ぶ．

局所制御では，実際に流れているトラヒック量を元に制御するため，現在流れているトラヒックに合わせた負荷分散が可能であり，ランダム制御よりも広い帯域をサーバー間に確保できると考えられる．また，各スイッチが自身の各ポートのキュー長などを元に，現在流れているトラヒック量をリアルタイムで把握することにより，トラヒック変動発生時にも瞬時にトラヒック変動に対応した負荷分散を行うことができる．しかし，局所制御では自身が接続しているリンクのトラヒック量は把握できるものの，他のスイッチに接続しているリンクのトラヒック量を把握せずに制御を行っているため，転送先のスイッチで輻輳が発生している場合もその輻輳を考慮した経路制御を行うことができず，サーバー間に十分な帯域を確保できなくなる可能性がある．

3.3 全体のトラヒック情報を用いる経路制御手法

データセンターネットワークの経路制御として，ネットワーク内の全地点間のトラヒック量を把握し，そのトラヒック量に合わせて経路を制御する手法も存在する [5]．この手法では，ネットワーク内のトラヒック量の把握・経路計算を行うサーバーを配置し，そのサーバーがネットワーク内の全機器からトラヒック量を収集，収集したトラヒック量に合わせた最適な経路を計算する．以降，この制御を全体制御と呼ぶ．

全体制御では，各時刻のトラヒックに合わせた最適な経路が設計されるため，トラヒック変動が起きない限り，ランダム制御や局所制御よりも大きな帯域をサーバー間に確保可能である．しかしながら，データセンターではトラヒック変動は頻繁に発生し，1 秒以内に大きくトラヒックの傾向が変わってしまう [5]．そのため，ネットワーク内の全機器から定期的な情報収集を行った後に経路を計算する全体制御では，トラヒック変動に追随できず，サーバー間に十分な帯域を確保することができない可能性がある．

4　データセンターにおけるネットワーク構成手法および経路制御手法の評価

4.1　評価の概要

本報告では，2 章で述べたデータセンターネットワークの構成と 3 章で述べた経路制御の組み合わせを評価し，データセンターネットワークに適したネットワーク構成と経路制御の組み合わせを明らかにする．

本評価では，評価対象ネットワーク構成上で評価対象の各経路制御手法を動作させ，経路制御手法で定められた経路にトラヒックの収容を行う．トラヒックは各サーバー間に発生するものとし，発生したトラヒックのうち，一定量のトラヒックずつ，各経路制御手法によって負荷が少なく，追加のトラヒックを収容する経路として適切だと判断された経路に収容を行う．そして，空帯域が存在する経路が経路制御手法により見つからず，さらなるトラヒックの収容ができなくなった時点で，トラヒックの収容を終了する．

上記の手順で，空帯域が存在する経路がなくなった時点で，各サーバー間を流れているトラヒック量が，各ネットワーク構成・経路制御手法の組み合わせにおいて，収容可能な最大のトラヒック量であると考えることができる．そこで，本評価は，空帯域が存在する経路がなくなった時点において，収容できたトラヒック量を調べることにより，適切な負荷分散を行うことができているのかを評価する．

また，データセンター内では機器の故障は頻繁に発生し，故障発生時もサービスを動作させ続ける必要がある．そこで，故障が発生した場合についても同様の評価も行い，故障発生時にもサーバー間に十分な帯域が確保可能なネットワーク構成と経路制御の組み合わせを明らかにする．

4.2　評価対象

4.2.1　ネットワーク構成

本評価では，各ネットワーク構成の環境を揃えるため，ポート数が等しいスイッチを用いた様々なネットワーク構成を生成し，比較を行う．比較対象のネットワーク構成として，本評価では，GFB のパラメーター設定によってポート数が等しいスイッチを同数用いた様々なネットワーク構成を生成した．GFB ではパラメーター L_k，M_k を変化させることにより，Torus など，ネットワーク内の全スイッチが同一の役割を持つ様々なネットワーク構成を構築することができる．

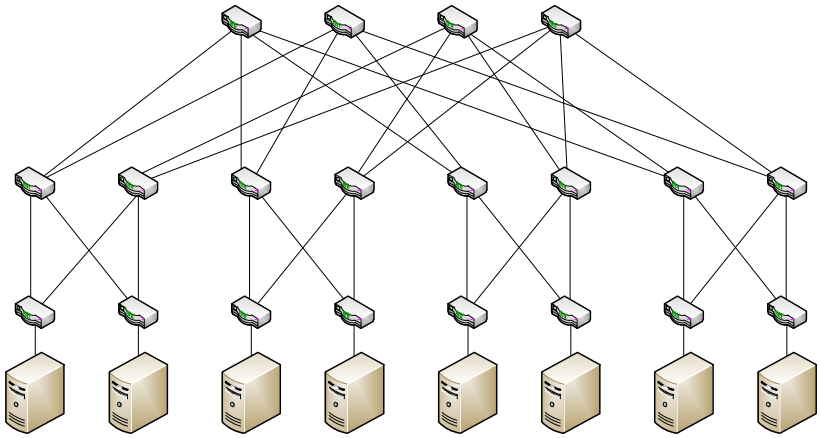


図 5: ポート数 4, 最下位層に配置するスイッチ数 8 の FatTree

しかしながら，GFB のパラメーターのみでは，スイッチが階層的に配置された木構造のネットワーク構成を生成することができない．そこで，本評価では GFB と合わせて，GFB に用いられたスイッチと同じポート数のスイッチを用いて構成される FatTree も用いる．FatTree では，ポート数 p ，階層数 k と最下位層に配置するスイッチ数 n の間で， $n = p(\frac{p}{2})^{k-2}(2 \leq k)$ の関係が成り立たないと，各スイッチの子ノードの数が偏ってしまう．図 5 はポート数 4 ，最下位層に配置するスイッチ数 8 で構成した全スイッチの子ノードの数が等しい FatTree を示し，図 6 はポート数 4 ，最下位層に配置するスイッチ数 7 で構成したスイッチの子ノードの数に偏りが生じている場合である．各スイッチの子ノードの数に偏りが生じた場合，スイッチによってリンク数が異なるため，リンク数が多いスイッチに負荷が集中するなど，十分な負荷分散を行うことができない可能性がある．そのため，本報告では， 子ノードの数に偏りが生じていない FatTree のみを評価対象とし，GFB で生成された他の評価対象ネットワーク構成と総スイッチ数が近い FatTree を生成した．

4.2.2 経路制御手法

本報告では，経路制御に用いられるトラヒック情報による影響に注目した評価を行う．そのため，ランダム制御，局所制御，全体制御のいずれの手法においても，各スイッチが選択しうる次ホップのスイッチの候補は等しくなるようにした．

ここで，転送先候補を定めるにあたり，以下のパラメータを導入する．

- 転送先候補保持ホップ数 h : 各スイッチは，受信先までのホップ数が最短ホップ数 $+h$ 以内にある経路情報を保持

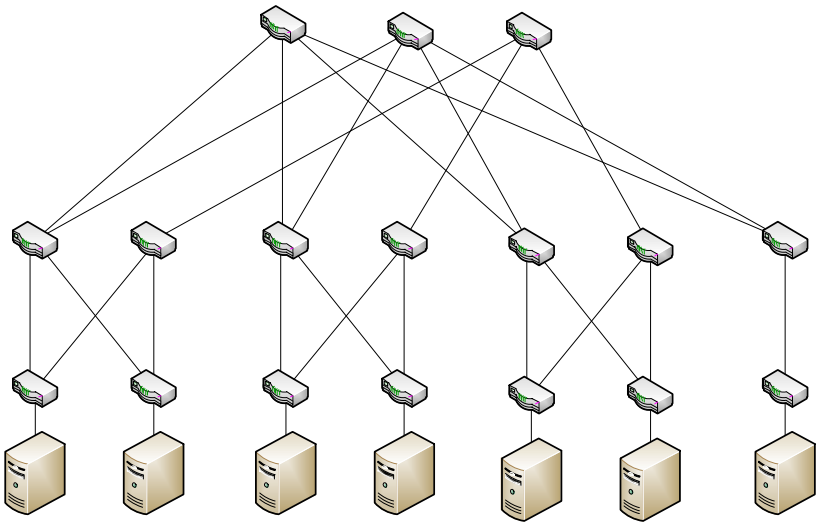


図 6: ポート数 4, 最下位層に配置するスイッチ数 7 の FatTree

- 許容ホップ数 s：送信元・受信先間の通信の際に，送信元・受信先間の最短ホップ数 $+s$ 以内のホップ数での通信を許容

h，s を変化させることにより，最短ホップのみを転送先の候補とした経路制御から，ホップ数が増大するような経路も転送先候補に含めた経路制御まで評価することができる．本評価では，パラメーター h によって定められた転送先の候補を各スイッチが保持し，パラメーター s によって，送信元・受信先間のホップ数を許容ホップ数以内に抑えることで，各制御で把握可能な情報に基づき，最も負荷が低い経路にトラヒックを収容する．

全体制御では，トラヒック情報をリアルタイムで把握することはできず，トラヒック変動に追随した制御を行うことができない可能性がある．本評価では，トラヒック変動後の影響も把握するため，全体制御において経路を確定する際に用いたトラヒックとは異なるトラヒックを生成した場合についても評価を行う．

さらに，本評価においては，故障発生時の影響についても評価を行う．故障が発生した際には，故障により，受信先まで到達することができなくなったスイッチを転送先の候補から除外する必要がある．これは，ネットワーク全体の情報を把握することなく，以下の手順によって可能である．まず，故障が発生したスイッチの隣接スイッチが，当該スイッチの故障の検知を行う．故障を検知したスイッチは，転送先候補から故障したスイッチを外す．そして，ある受信先までの転送先候補となるスイッチがなくなったにも関わらず，当該受信先スイッチ宛のパケットを受信した場合は，1 ホップ前のスイッチに当該パケットを送り返す．これにより，1 ホップ前のスイッチにおいても，パケット返送元のスイッチを経由した受信先

までの経路が存在しないことを知ることができ，転送先の候補から除外することができる．この手順を繰り返すことにより，全スイッチの転送先候補から，受信先まで転送できない転送先候補を除去することができる．

本評価においては，上記の手順により，故障発生により経路が存在しない転送先の候補を除外した上で，各経路制御手法を動作させた場合について評価を行う．そして，故障発生時にもサーバー間に大きな帯域を確保できる，データセンターネットワークに適したネットワーク構成と経路制御手法を明らかにする．

4.3 評価環境

4.3.1 トラヒック発生方法

データセンター内には，小さなメッセージをサーバー間で交換する際に発生する小さなトラヒックと，ファイルなどの大きなデータをサーバー間で交換する際に生じる大きなトラヒックが存在すると考えられる．

以降，本報告では小さなメッセージの交換により生じる小さなトラヒックを軽量トラヒックと呼び，ファイルなどの大きなデータをサーバー間で交換する際に生じる大きなトラヒックを帯域圧迫トラヒックと呼ぶ．評価の際には，各送信元・受信先に流れるトラヒック量は，軽量トラヒックと帯域圧迫トラヒックの合計であるとする．

本評価において，軽量トラヒックは，全サーバー間に発生しているものとし，そのトラヒック量は一様乱数で生成した．帯域圧迫トラヒックは，一部のサーバー間のみで発生しているものとし，各スイッチあたり 1 本の割合で，ランダムに選択した送信元・受信先に帯域圧迫トラヒックを生成した．帯域圧迫トラヒックが流れるサーバー間では，可能な限り多くのデータを送信しようとしている場面を想定し，トラヒックが収容される経路に空帯域がある限り大きなトラヒックを発生させた．

4.3.2 故障発生方法

本評価では，リンクやスイッチの故障を想定する．リンク・スイッチのいずれの故障においても，故障の発生はランダムであるとした．

4.4 評価指標

本評価では，以下の指標を用いることにより，負荷分散により多くのトラヒックを収容可能なデータセンターに適したネットワーク構成と経路制御手法の組み合わせを明らかにする．

4.4.1 送信可能なトラヒックの合計

各送信元・受信先間に送信可能なトラヒックの合計を評価指標として用いることにより，ネットワークが収容することができるトラヒックの総量を求めることができ，より多くのトラヒックを収容可能なネットワーク構成・経路制御手法を明らかにすることができる．

4.4.2 帯域圧迫トラヒックの最小値

帯域圧迫トラヒックの最小値は，帯域圧迫トラヒックを流す送信元・受信先間のうち，送信可能なトラヒック量が最も少ない送信元・受信先間で送信できたトラヒック量を示す指標である．帯域圧迫トラヒックの最小値を指標として用いることにより，各スイッチ間に確保可能な通信帯域の最小値を得ることができ，いずれのサーバー間にも大きな帯域を確保可能なネットワーク構成・経路制御手法を明らかにすることができる．

4.4.3 平均ホップ数

平均ホップ数は各送信元・受信先間のトラヒックが通過するスイッチ数の平均値である．平均ホップ数を評価することにより，各送信元・受信先間のトラヒックにより帯域が消費されるリンクの本数を得ることができ，本報告では，それを元により多くのトラヒックを収容可能なネットワーク構成・経路制御手法に関する考察を行う．

4.4.4 通信の失敗率

本報告では，故障が発生した際に，全送信元・受信先間の組み合わせ数に対して，動作させた経路制御手法によって到達可能な経路を見つけることができない送信元・受信先の組み合わせの割合を通信の失敗率と呼ぶ．

通信の失敗率が小さいネットワーク構成・経路制御手法は，故障発生時にも通信が可能なサーバー間が多く，故障に強いと言える．

5　評価結果

本章では，まず，トラヒック変動の経路制御への影響を調べる．その後，故障が発生しない条件において，サーバー間の多量の通信を収容可能な適切なネットワーク構成・経路制御手法について議論する．そして，最後に，故障が発生した場合にもサーバー間に通信帯域を確保可能な適切なネットワーク構成・経路制御手法について議論する．

5.1　トラヒック変動の経路制御手法への影響

トラヒック変動が発生した場合においても，トラヒック情報を用いず制御を行っているランダム制御や，各スイッチがポートごとのキュー長などの情報をリアルタイムで把握して制御を行う局所制御では，トラヒック変動の影響を受けず，現在のトラヒックに合わせた制御を行うことができる．しかしながら，収集したトラヒック情報に基づいて制御を行う全体制御では，トラヒック変動後のトラヒック情報をすぐに把握することはできず，以前のトラヒック情報に基づいて設計された経路にトラヒックが収容される．本小節では，ランダムに生成した初期トラヒック状況から著しくトラヒックが変動した状態を，新たに軽量トラヒックの生成，帯域圧迫トラヒックの送受信スイッチの選択を行うことにより生成し，トラヒック変動が発生した直後にネットワーク内に収容されるトラヒックについて評価を行う．そして，トラヒック変動が経路制御へ与える影響を議論する．

本小節の評価においては，表 1，2 のネットワーク構成を用い，各ネットワーク構成において，全体制御，局所制御，ランダム制御で経路制御が行われた際に，サーバー間に送信可能なトラヒックの合計や，帯域圧迫トラヒックが発生しているスイッチ間で送信可能なトラヒックの最小値を比較する．また，合わせて，トラヒック変動後の正確なトラヒック情報を用いて全体制御を行った場合の理想的な全体制御との比較も行う．

図 7 に送信可能なトラヒックの合計，図 8 に帯域圧迫トラヒックの最小値の結果を示す．図 7 より，どのネットワーク構成においても，全体制御は，正確なトラヒック情報を把握可能な理想的な場合は最も多くのトラヒックを送信可能であるものの，トラヒック変動が起きた直後に過去のトラヒック状況に対して設計された経路をそのまま用いた場合に送信可能なトラヒックの合計は，ランダム制御と同程度の値となっている．また，同様に，図 8 においても，トラヒック変動が起きた直後に過去のトラヒック情報に対して設計された経路をそのまま用いた場合に送信可能な帯域圧迫トラヒックの最小値は，ランダム制御と同程度，局所制御よりも少ない値となっている．これは，全体制御で経路を設計する際に用いたトラヒック情報とは異なる箇所に帯域圧迫トラヒックが発生していることが原因である．本報告で用いた全体制御ではネットワークのトラヒック情報を元に，空帯域がある経路に送受信間の

表 1: スイッチ数 120，ポート数 6 の GFB の構成

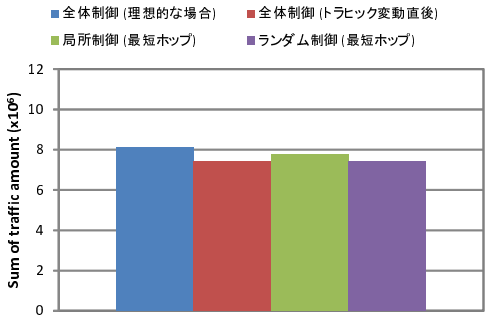
識別名	N_0	N_1	N_2	L_0	L_1	L_2	M_0	M_1	M_2	全スイッチ数	リンク数
GFB213	4	5	6	2	1	3	1	1	1	120	360
GFB222	4	5	6	2	2	2	1	1	1	120	360
GFB231	4	5	6	2	3	1	1	1	1	120	360
GFB312	4	5	6	3	1	2	1	1	1	120	360
GFB321	4	5	6	3	2	1	1	1	1	120	360
Torus	4	5	6	2	2	2	1	4	20	120	360

表 2: ポート数 6 の FatTree の構成

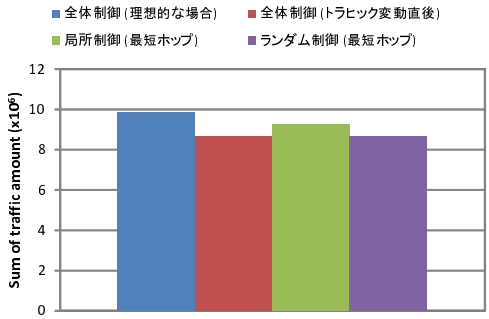
識別名	最下位層のスイッチ数	ポート数	階層数	全スイッチ数	リンク数
FatTree (3 layers)	18	6	3	47	108
FatTree (4 layers)	54	6	4	189	486

トラヒックを収容することを繰り返すことにより，帯域圧迫トラヒックのような多量のトラヒックの収容を行っている．しかしながら，帯域圧迫トラヒックが発生するスイッチ間が変化した場合には，トラヒック変動により新たに帯域圧迫トラヒックが流れるようになったスイッチ間には，空帯域を有効利用するような経路設定は行われていない．その結果，ランダム制御と同様，現在のトラヒック状況を考慮せず，過去のトラヒック情報に合わせて設計された確率に従って，トラヒックの収容先の候補が決められる．

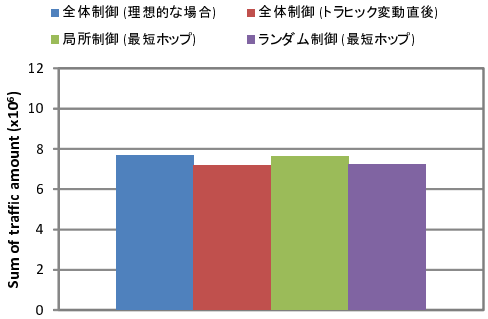
このように，全体制御では，現在のトラヒック情報を把握することができない際には，収容可能なトラヒック量の著しい悪化を招く．また，データセンター内では，著しいトラヒック変動が発生する [5]．そのため，文献 [5] において議論されているように，全体制御において，現在のトラヒックに対して適切な経路を維持するためには，1 秒間隔で経路の最適化を行う必要がある．しかしながら，データセンターの規模が大きくなると，1 秒単位でネットワーク全体のトラヒック量を観測し，経路の再設計を行うのは困難である．そのため，環境変動の激しいデータセンターネットワーク内においては，ネットワーク全体の情報を用いた制御ではなく，局所的なトラヒック情報を用いた経路制御が望ましく，局所的なトラヒック情報を用いた経路制御が適切に動作するネットワーク構成が望ましいと考えられる．



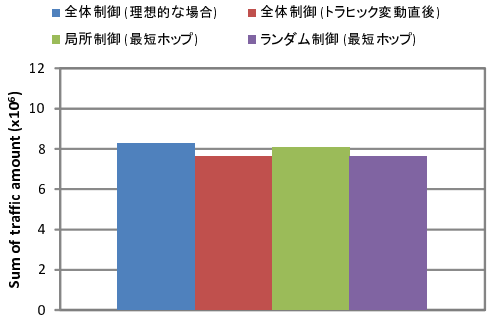
(a) GFB213



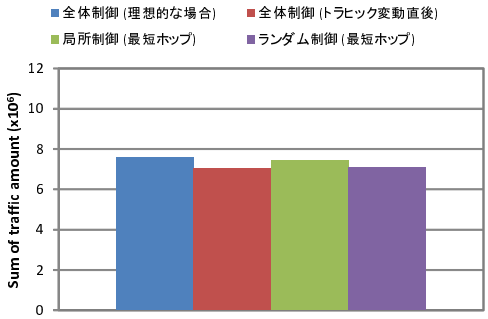
(b) GFB222



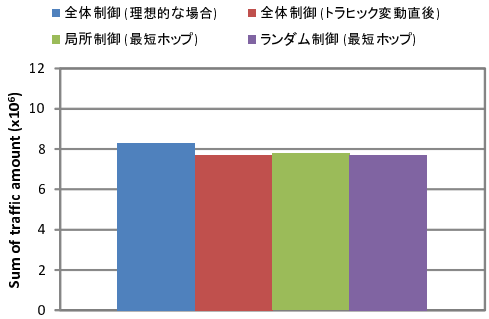
(c) GFB231



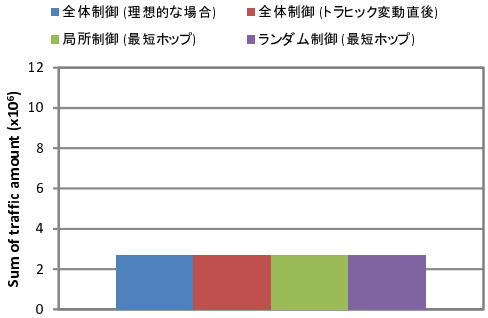
(d) GFB312



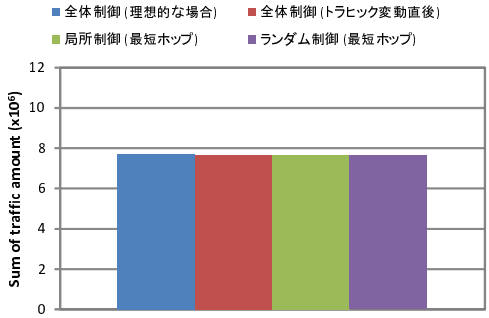
(e) GFB321



(f) torus

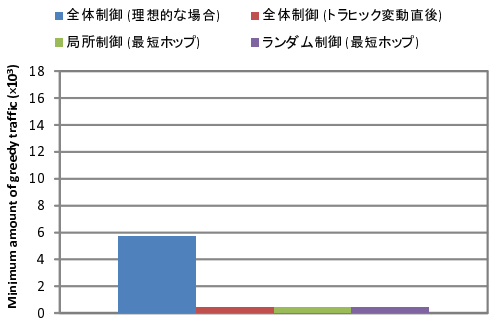


(g) 3 階層の FatTree

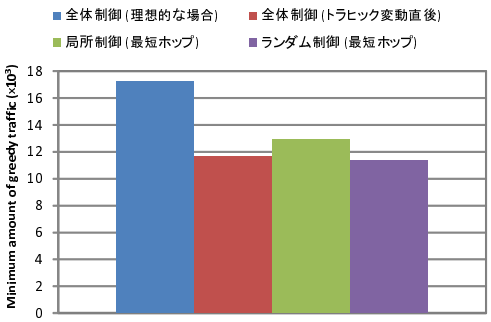


(h) 4 階層の FatTree

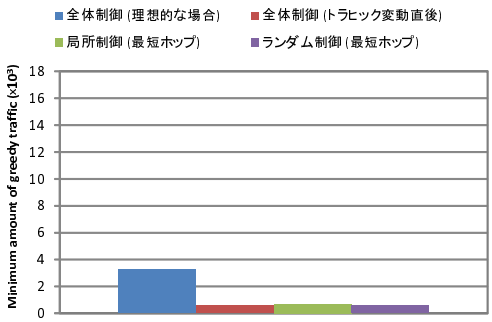
図 7: トラヒック変動発生直後の送信可能なトラヒックの合計



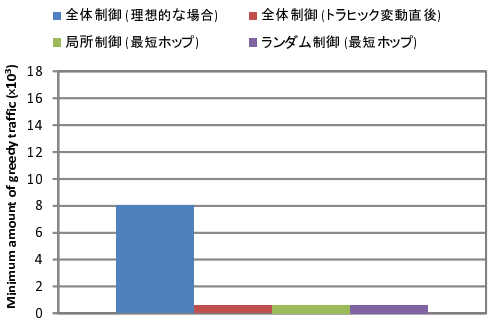
(a) GFB213



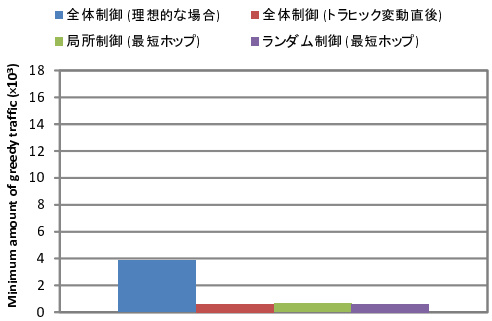
(b) GFB222



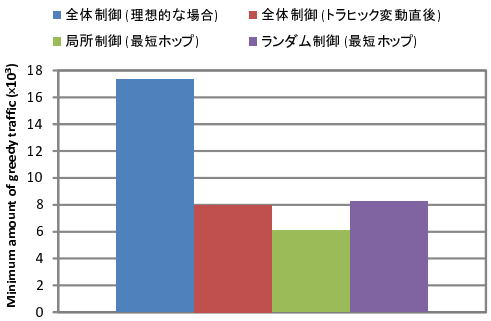
(c) GFB231



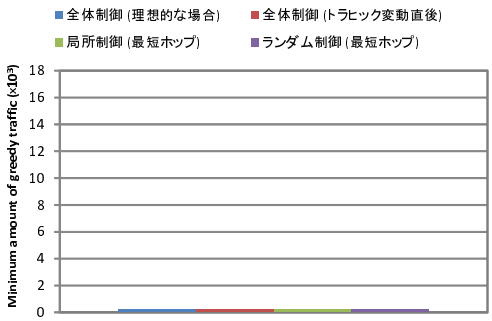
(d) GFB312



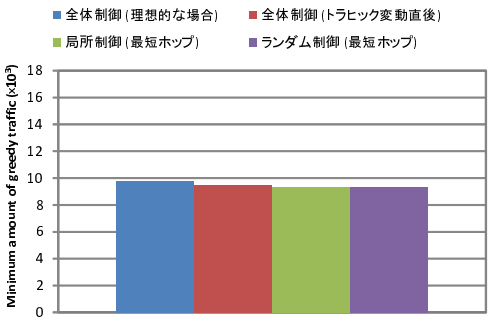
(e) GFB321



(f) torus



(g) 3 階層の FatTree



(h) 4 階層の FatTree

図 8: トラヒック変動発生直後の帯域圧迫トラヒックの最小値

5.2 負荷分散により多量のトラヒックを収容可能なネットワーク構成

本小節では，ポート数が 6，あるいは 8 のスイッチを 120 台，あるいは 220 台用いることにより構成可能なネットワーク上で，評価対象の経路制御手法を動作させて，多くのトラヒックを収容可能なネットワーク構成と経路制御の組み合わせを明らかにする．本小節の評価においては，局所制御，ランダム制御ともに，最短ホップ経路のみをトラヒック転送先の経路として考えた場合と，最短ホップよりも 1 ホップ多い経路までトラヒックの転送先として考えた場合を評価対象とする．また，比較対象として，ネットワーク全体の正確なトラヒック量が把握できた場合に全体制御を行った場合も評価する．

5.2.1 ポート数 6 のスイッチを 120 台用いたネットワーク構成の比較

表 1 に示されるポート数 6 のスイッチ 120 台で構成可能なネットワークの評価を行った．また，120 台のスイッチを用いて全てのスイッチの子ノードの数が等しくなる FatTree を構成することはできないため，表 2 に示されるように，6 ポートのスイッチを用いて構成されたスイッチ数が 120 台よりも少ない 3 階層の FatTree，120 台よりも多い 4 階層の FatTree を用いた．

図 9 にネットワーク内で送信可能なトラヒックの合計を示す．図より，いずれの経路制御手法においても，GFB222 が最も送信可能なトラヒックの合計が大きい．これは，GFB213，GFB231，GFB312，GFB321 のように，各スイッチあたりのリンク数が 1 となっている階層が存在する場合は，リンク数が少ない階層で輻輳が発生し，多くのトラヒックを収容することができないためである．

3 階層の FatTree は，送信可能なトラヒックの合計が他と比べて著しく小さい．これは，ネットワーク内のリンク数が少ないためであると考えられる．しかしながら，ネットワーク内のリンク数が多い 4 階層の FatTree も送信可能なトラヒックの合計は GFB222 よりも少ない．これは，FatTree においては，図 10 に示されるように，平均ホップ数が大きくなっていることが原因である．FatTree では上位の階層を経由する通信が多いため，ホップ数が大きくなる．その結果，ホップ数が大きい通信は多数のリンクの帯域を消費し，送信可能なトラヒックの合計が小さくなっている．

FatTree と同様，Torus も図 10 に示されるように，平均ホップ数が大きく，ホップ数の大きい通信が多数のリンクの帯域を消費するため，GFB222 よりも，送信可能なトラヒックの合計が小さくなっている．

図 11 に帯域圧迫トラヒックの最小値の比較の結果を示す．図 11 より，送信可能なトラヒックの合計と同様，GFB222 が帯域圧迫トラヒックの最小値が最も大きいことがわかる．GFB222 は局所制御においても，帯域圧迫トラヒック発生スイッチ間には，10^4 以上の帯域

を確保できているのに対して，GFB213，GFB231，GFB312，GFB321 は，いずれも局所制御では，帯域をほとんど確保することができないスイッチ間が存在していることがわかる．これは，各スイッチから 1 本しかリンクが構築されていない階層において，輻輳が発生することが原因である．各スイッチから 1 本しかリンクが構築されていない階層において輻輳が発生した場合も，全体制御では輻輳が発生したリンクを避けるような経路を選択することができるものの，局所制御では各スイッチが転送先のスイッチを選択する際には，転送先のスイッチで発生している輻輳を把握することができない．さらに，輻輳が発生したリンクを持つスイッチにおいても，当該階層のリンクは輻輳が発生したリンクのみであり，輻輳を迂回する経路を選択することができない．その結果，帯域圧迫トラヒックが発生している送信元・受信間の位置によっては十分な帯域を確保可能な経路を見つけることができず，送信可能な帯域圧迫トラヒックの最小値が小さくなっている．

4 階層の FatTree，Torus は，局所制御やランダム制御においても，帯域圧迫トラヒック発生スイッチ間に 5.0×10^3 以上の通信帯域を確保可能である．これは，いずれのネットワーク構成においても，階層間や各次元に同数のリンクが割り当てられており，局所的な情報を用いても輻輳の回避ができる転送先を選択可能であるためである．しかしながら，これらのネットワーク構成はホップ数が大きく，一つのスイッチ間のトラヒックが多くのリンクの帯域を消費してしまうため，GFB222 ほど多くのトラヒックを収容することができない．

また，図 11 から，最短ホップ以外の経路も転送先の候補とした局所制御を行うことが GFB222 においては，有効であることがわかる．これは，最短ホップ以外の経路も候補として保持することにより最短ホップの経路のみを考えた転送先リンクが輻輳している場合でも，輻輳しているリンクを迂回する経路を各スイッチが把握している局所的な情報から選択可能となるためである．

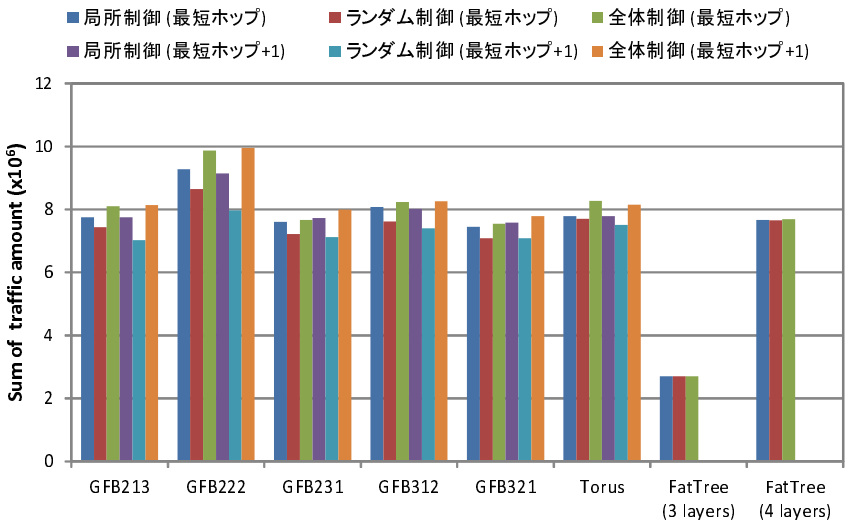


図 9: 6 ポートのスイッチ 120 台を用いたネットワーク構成における送信可能なトラヒックの合計

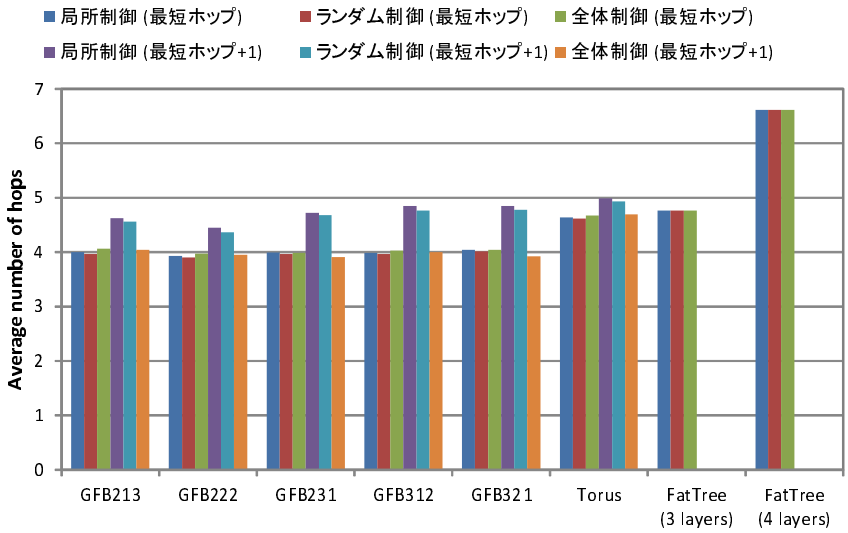


図 10: 6 ポートのスイッチ 120 台を用いたネットワーク構成における平均ホップ数

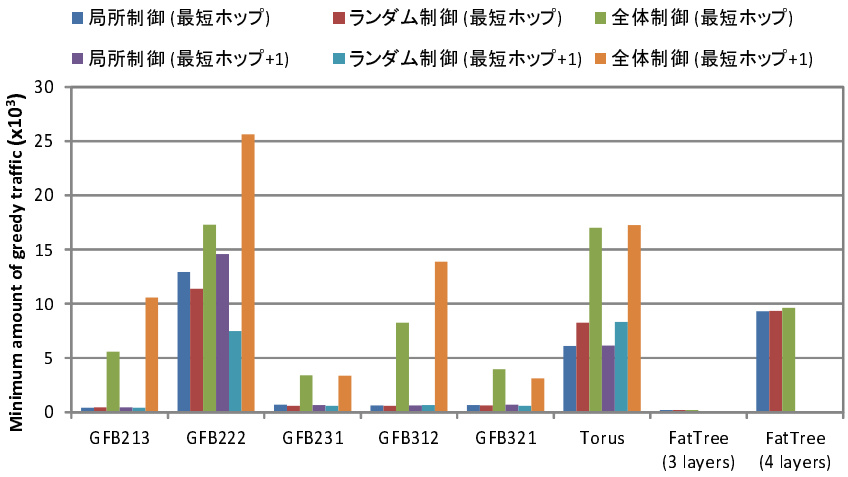


図 11: 6 ポートのスイッチ 120 台を用いたネットワーク構成における帯域圧迫トラヒックの最小値

5.2.2 ポート数 6 のスイッチを 220 台用いたネットワーク構成の比較

表 3 に示されるポート数 6 のスイッチ 220 台で構成可能なネットワークの評価を行った．また，Torus については，各次元のスイッチ数 N_k の値を他の GFB と同じ値を用いた Torus $(4 \times 5 \times 11)$ と，各次元のスイッチ数を 6 に統一した Torus $(6 \times 6 \times 6)$ を用いた．

図 13 は送信可能なトラヒックの合計を示す．図 13 より，120 台のスイッチで構築したネットワーク構成の比較の結果と同様，GFB222 がいずれの経路制御においても，最も送信可能なトラヒックの合計が大きいことがわかる．表 1 と異なり，表 3 の GFB では，3 階層の GFB の数が多いにも関わらず，3 階層の GFB 間のリンク数が多い GFB213 よりも GFB222 が送信可能なトラヒックの合計が大きい．これは，GFB213 では，各スイッチあたり 1 本しかリンクがない階層が存在し，その階層内で発生した輻輳を迂回する経路が十分に存在しないことが原因である．

また，Torus で送信可能なトラヒックの合計も 120 台のスイッチで構築したネットワーク構成の比較の結果と同様，GFB222 より少なくなっている．これは，GFB222 よりも Torus 内のスイッチ間の平均ホップ数が大きいため，スイッチ間のトラヒックが多くのリンクの帯域を消費するためである．Torus $(4 \times 5 \times 11)$ と，Torus $(6 \times 6 \times 6)$ を比較すると，Torus $(6 \times 6 \times 6)$ の方が送信可能なトラヒック量が多い．これは，Torus ではスイッチ数の多い次元が存在すると，ホップ数が大きくなるため，Torus $(4 \times 5 \times 11)$ は Torus $(6 \times 6 \times 6)$ よりも平均ホップ数が大きく，各スイッチ間トラヒックが多くのリンクの帯域を消費してしまうためである．

GFB222 における局所制御では，最短ホップよりも 1 ホップ多い経路も候補とした制御を行った場合に送信可能なトラヒックの合計は，最短ホップのみを候補とした制御よりも少なくなっている．これは，最短ホップよりも 1 ホップ多い経路も候補とすることで，スイッチ間のトラヒックが経由するホップ数が増えてしまい，より多くのリンクの帯域を消費するようになるためである．220 台のスイッチを接続した環境では，120 台のスイッチを接続した環境よりも，トラヒックが発生するスイッチ間の組み合わせ数に対して，リンク数が少ない．そのため，ホップ数の増加による帯域が消費されるリンク数の増加が，送信可能なトラヒック量に与える影響が大きい．つまり，構築されたリンク数が少ないネットワークにおいて，送信可能なトラヒックの合計を増やすためには，ホップ数を抑える経路制御を行うことが重要であると考えられる．

図 12 に帯域圧迫トラヒックの最小値を示す．図 13 より，Torus $(6 \times 6 \times 6)$ の帯域圧迫トラヒックの最小値は大きく，特に局所制御における帯域圧迫トラヒックの最小値は，評価を行ったネットワーク構成の中で最大である．Torus $(6 \times 6 \times 6)$ では，各次元のスイッチの数が等しく，スイッチ間に同一ホップ数の経路が多い．そのため，輻輳が発生したリンクが生

表 3: スイッチ数 220，ポート数 6 の GFB と Torus の構成

識別名	N_0	N_1	N_2	L_0	L_1	L_2	M_0	M_1	M_2	全スイッチ数	リンク数
GFB213	4	5	11	2	1	3	1	1	1	220	660
GFB222	4	5	11	2	2	2	1	1	1	220	660
GFB231	4	5	11	2	3	1	1	1	1	220	660
GFB312	4	5	11	3	1	2	1	1	1	220	660
GFB321	4	5	11	3	2	1	1	1	1	220	660
Torus $(4 \times 5 \times 11)$	4	5	11	2	2	2	1	4	20	220	660
Torus $(6 \times 6 \times 6)$	6	6	6	2	2	2	1	6	36	216	648

じた場合も，局所的な制御により別の経路の選択が可能であり，帯域圧迫トラヒックを送受信しているスイッチ間に大きな帯域を確保することが可能である．

GFB222 は，全体制御では帯域圧迫トラヒックの最小値は最も大きいものの，局所制御を行った場合の帯域圧迫トラヒックの最小値は Torus $(6 \times 6 \times 6)$ よりも小さくなる．これは，GFB222 は，Torus ほどスイッチ間の最短ホップの経路の本数が多くなく，各スイッチが転送先として選択できる候補の数は少ないためである．特に，接続している GFB の数に対してリンク数が少ない 3 層のリンクで輻輳が発生しやすく，各スイッチにおいて，その発生した輻輳を解消する経路を局所的な情報から見つけることはできず，送信可能なトラヒック量は制限されてしまう．しかしながら，局所制御を行った場合においても，GFB222 における帯域圧迫トラヒックの最小値と Torus の帯域圧迫トラヒックの最小値の差は少なく，GFB222 もスイッチ間に大きな帯域を確保することができる構成であると考えることができる．

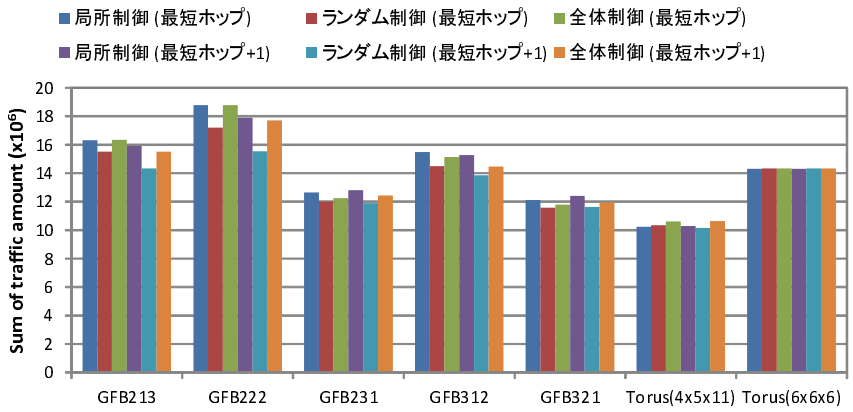


図 12: 6 ポートのスイッチ 220 台を用いたネットワーク構成における送信可能なトラヒックの合計

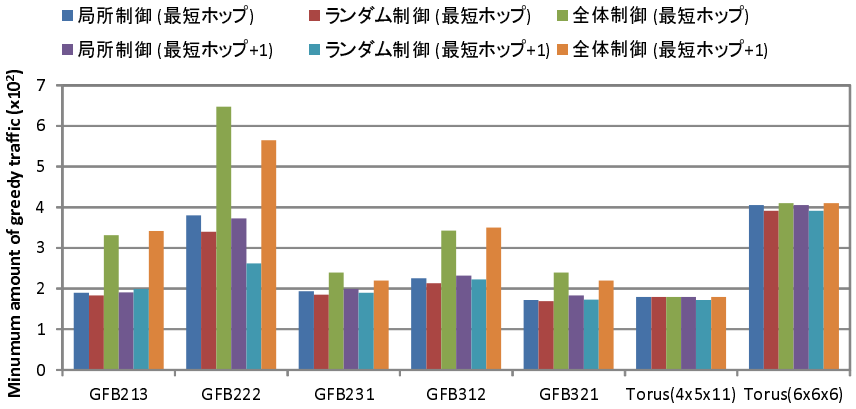


図 13: 6 ポートのスイッチ 220 台を用いたネットワーク構成における帯域圧迫トラヒックの最小値

5.2.3　ポート数 8 のスイッチを 120 台用いたネットワーク構成の比較

表 4,5 に示されるポート数 8 のスイッチを 120 台用いた GFB と，ポート数 8 のスイッチを 135 台用いた 4 次元 Torus を用いた評価を行った．

図 14 は送信可能なトラヒックの合計を，図 15 は帯域圧迫トラヒックの最小値を表している．いずれの結果も，ポート数が 6 のスイッチを用いた評価と同様，各スイッチあたりのリンク数が 1 の階層が存在する構成では，送信可能なトラヒックの合計，帯域圧迫トラヒックの最小値ともに小さくなっている．また，本評価で用いた Torus はスイッチ数が 135 と他のネットワーク構成よりもスイッチ数が多いにも関わらず，送信可能なトラヒックの合計は他のネットワーク構成と同程度，帯域圧迫トラヒックの最小値は他のネットワーク構成よりも小さな値となっている．これは，Torus では平均ホップ数が大きいため，各スイッチ間のトラヒックが多くのリンクの帯域を消費してしまうためである．

各スイッチあたりのリンク数が 1 の階層が存在しない構成の中でも，GFB233 が最も多くのトラヒックを送信可能である．これは，GFB233 では，局所的な情報を用いた経路制御で選択可能な各階層の複数の経路を確保しつつ，輻輳が発生しやすい上位層により多くのリンクを割りあてることができているためであると考えられる．GFB332 や GFB323 のような接続している GFB の数が少ない最下位層に多くのリンクを割り当てるよりも，接続している GFB の数が多く輻輳が発生しやすい上位層により多くのリンクを割り当てた方が，輻輳を発生させず，より多くのトラヒックを収容可能である．また，GFB224 のように最上位層により多くのリンクを構築した構造よりも，各スイッチが各階層に用いるリンク数が均等な GFB233 の方が，第二層，最上位層のいずれで輻輳が発生した場合にも，局所的な情報のみで選択可能な迂回経路が複数存在するため，局所制御での輻輳の解消が容易である．

また，最短ホップのみを転送先の候補とした経路制御よりも，最短ホップより 1 ホップ多い経路も転送先の候補に含めた経路制御が，より多くのトラヒックを収容できている．これは，8 ポートのスイッチを用いた場合は，6 ポートのスイッチを用いて構成した場合よりもリンク数が多く，帯域圧迫トラヒックが発生しているスイッチ間の最短ホップ経路に含まれないリンクが存在するためであると考えられる．そして，最短ホップ以外の経路も考慮することにより，そのような最短ホップ経路に含まれないリンクの帯域も有効に活用することができるようになり，より多くのトラヒックを流すことができるようになる．

表 4: スイッチ数 120，ポート数 8 の GFB の構成

識別名	N_0	N_1	N_2	L_0	L_1	L_2	M_0	M_1	M_2	全スイッチ数	リンク数
GFB215	4	5	6	2	1	5	1	1	1	120	480
GFB224	4	5	6	2	2	4	1	1	1	120	480
GFB233	4	5	6	2	3	3	1	1	1	120	480
GFB242	4	5	6	2	4	2	1	1	1	120	480
GFB251	4	5	6	2	5	1	1	1	1	120	480
GFB314	4	5	6	3	1	4	1	1	1	120	480
GFB323	4	5	6	3	2	3	1	1	1	120	480
GFB332	4	5	6	3	3	2	1	1	1	120	480
GFB341	4	5	6	3	4	1	1	1	1	120	480

表 5: スイッチ数 135，ポート数 8 の Torus の構成

識別名	N_0	N_1	N_2	N_3	L_0	L_1	L_2	L_3	M_0	M_1	M_2	N_3	全スイッチ数	リンク数
Torus	3	3	3	5	2	2	2	2	1	3	9	27	135	540

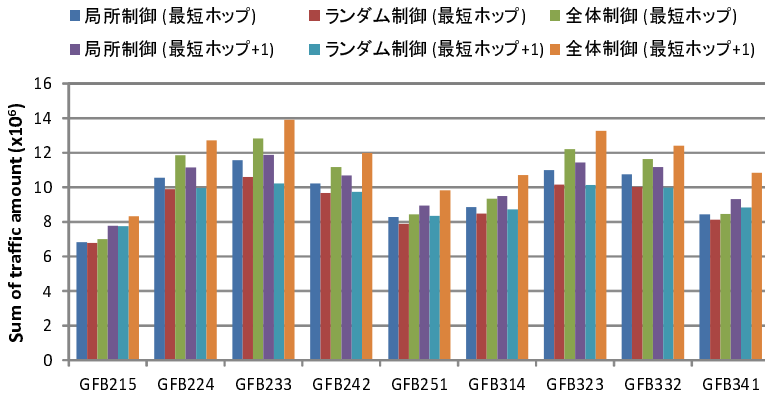


図 14: 8 ポートのスイッチ 120 台を用いたネットワーク構成における各送信元・受信先間での送信可能なトラヒックの合計

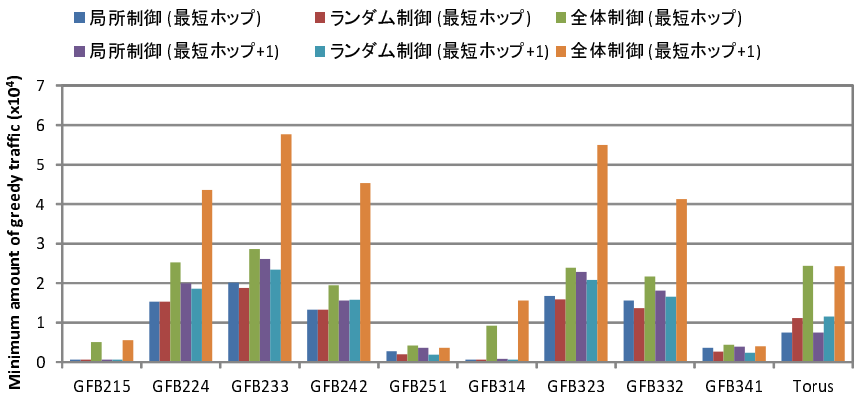


図 15: 8 ポートのスイッチ 120 台を用いたネットワーク構成における帯域圧迫トラヒックの最小値

表 6: スイッチ数 220，ポート数 8 の GFB の構成

識別名	N_0	N_1	N_2	L_0	L_1	L_2	M_0	M_1	M_2	全スイッチ数	リンク数
GFB215	4	5	11	2	1	5	1	1	1	220	880
GFB224	4	5	11	2	2	4	1	1	1	220	880
GFB233	4	5	11	2	3	3	1	1	1	220	880
GFB242	4	5	11	2	4	2	1	1	1	220	880
GFB251	4	5	11	2	5	1	1	1	1	220	880
GFB314	4	5	11	3	1	4	1	1	1	220	880
GFB323	4	5	11	3	2	3	1	1	1	220	880
GFB332	4	5	11	3	3	2	1	1	1	220	880
GFB341	4	5	11	3	4	1	1	1	1	220	880

5.2.4 ポート数 8 のスイッチを 220 台用いたネットワーク構成の比較

表 6，7 に示されるポート数 8 のスイッチを 220 台用いた GFB と，ポート数 8 のスイッチを 225 台用いた 4 次元 Torus を用いて評価を行った．

図 16 に送信可能なトラヒックの合計を示す．図 16 より，GFB224，GFB233 といった相互接続する GFB の数が多い上位層に多くのリンクが構築されている構造が，より多くのトラヒックを収容できることがわかる．これは，より多くの GFB が相互接続されている上位層により多くのリンクを割り当てることにより，GFB 間に十分な帯域を確保することができるためである．

図 17 に帯域圧迫トラヒックの最小値を示す．図 17 より，局所制御を用いた場合，最も帯域圧迫トラヒックの最小値を大きくすることができるのは，最短ホップより 1 ホップ多い経路まで候補として扱った場合の GFB233 である．GFB224 は，他の層よりも相互接続される GFB の数が著しく多い第 3 層により多くのリンクを割り当てているにも関わらず，局所制御を行った場合の帯域圧迫トラヒックの最小値は GFB233 よりも小さい．これは，GFB224 のような各スイッチあたりのリンク数が特定の層のみ多いよりは，全階層で均等な数のリンクを用いた方が，いずれの層で発生した輻輳にも局所的な情報のみで迂回経路を発見することができるためであると考えられる．そのため，スイッチ数によらず，局所制御によってサーバ間に十分な帯域を確保するためには，各スイッチあたりの各階層のリンク数は同程度となる構造が望ましいと考えられる．

表 7: スイッチ数 225，ポート数 8 の Torus の構成

識別名	N_0	N_1	N_2	N_3	L_0	L_1	L_2	L_3	M_0	M_1	M_2	N_3	全スイッチ数	リンク数
Torus	3	3	5	5	2	2	2	2	1	2	9	45	225	890

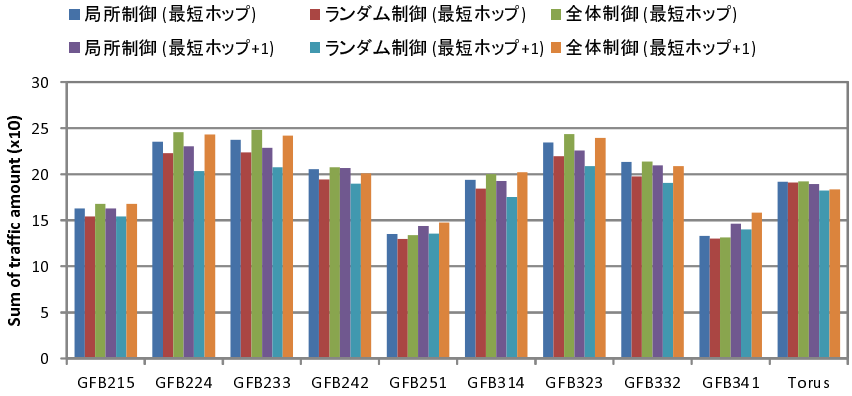


図 16: 8 ポートのスイッチ 220 台を用いたネットワーク構成における各送信元・受信先間での送信可能なトラヒックの合計

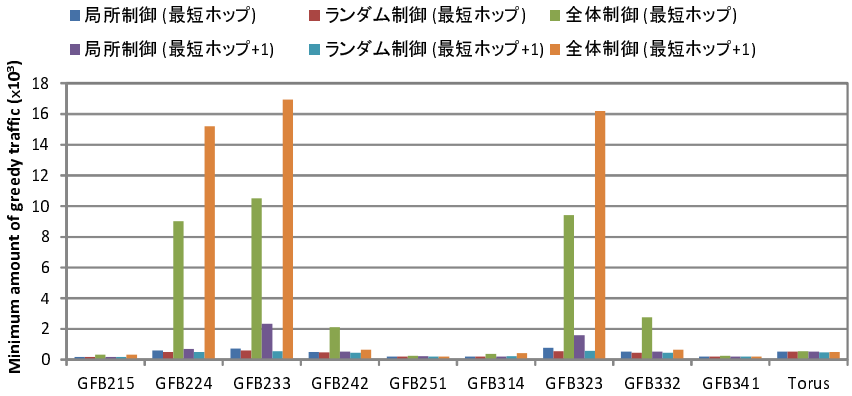


図 17: 8 ポートのスイッチ 220 台を用いたネットワーク構成における帯域圧迫トラヒックの最小値

5.2.5 まとめ

多くのトラヒックを収容するためには，ホップ数が大きく，特定のスイッチ間のトラヒックが多数のリンクの帯域を消費するような構成は避ける必要があり，GFB のように階層的な構造により，平均ホップ数を小さくした構造が適していると考えられる．

また，GFB のように階層的に構築した構造において，各スイッチあたりの各階層の接続に用いるリンク数は同程度とすることにより，いずれの層にトラヒックが集中した場合であっても，各スイッチが局所的な情報を元に輻輳箇所を迂回可能な転送先を発見することができるようになる．そのため，局所制御によってサーバー間に十分な通信帯域を確保するためには，各スイッチあたりの各階層の接続に用いられるリンク数は均等とする必要があると考えられる．

また，最短ホップ以外の経路も候補として含めた経路制御は，サーバー間に確保可能な最小の帯域の向上や，発生した帯域圧迫トラヒックが少なく，最短ホップで経路計算されると帯域圧迫トラヒックが経由しないリンクが存在するような場合の，送信可能トラヒックの合計の向上に有効である．

5.3 故障に対応可能なネットワーク構成と経路制御手法

本節では，リンクの故障およびスイッチの故障によってネットワークに与える影響の評価を行う．評価対象のネットワーク構成として，表 1，2 を用いる．

5.3.1 リンクに故障が発生した場合

まず，リンク故障が与える影響として，通信の失敗率の評価を行った．ここでは，局所的な情報を用いた転送先の決定を想定し，故障発生後に新たな経路計算を行わず，各スイッチが把握している転送先候補のうち，受信先まで到達できない候補を除外することのみを行った場合に，送信元から受信先までトラヒックを届けることができない割合を評価した．

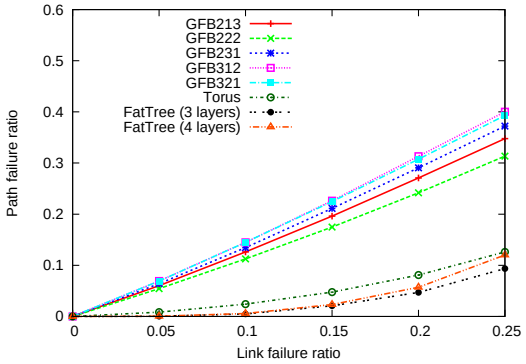
図 18(a) は最短ホップの経路のみ各スイッチが転送先の候補として保持している場合，図 18(b) は最短ホップ経路よりも 1 ホップ多い経路を転送先の候補として保持している場合，図 18(c) は全ての経路を転送先の候補としている場合の，リンク故障率に対する送信元・受信先間の通信の失敗率である．

図 18(a) から FatTree および Torus は，最短ホップ経路のみを転送先候補として保持する経路制御においても通信失敗率が小さいのに対して，他の GFB のネットワーク構成では最短ホップ経路のみを転送先の候補とした場合の通信失敗率が大きいことがわかる．GFB では，ホップ数が小さくなるように階層的な構造をとっており，各スイッチ間のトラヒックが経由する最短ホップ経路の種類が少ない．その結果，故障リンク数が多くなくても，全ての最短ホップ経路が故障により切断される送信元・受信先間が発生する．それに対して FatTree や Torus では，スイッチ間に多数の最短ホップ経路が存在する．そのため，最短ホップの経路のみを候補として各スイッチが保持している場合であっても，通信の失敗率は少なくなる．

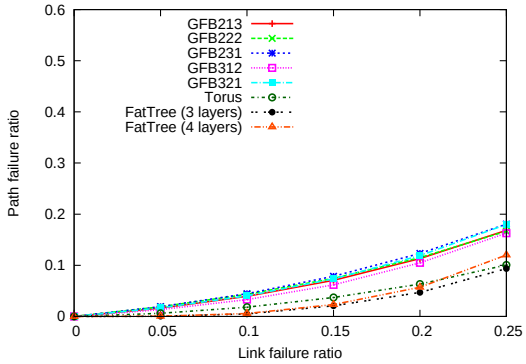
しかしながら，図 18(b) より，最短ホップより 1 ホップ多い経路も候補として保持することにより，GFB の通信失敗率が小さくすることが可能であることがわかる．また，図 21(c) から，全ての経路を考慮した場合は，通信が失敗する送信元・受信先間の割合は著しく小さい．以上のことから，GFB においても，最短ホップ以外の経路も考慮した経路制御を行うことで故障箇所を迂回できることがわかる．

図 19 は最短ホップ経路のみを，図 20 は最短ホップより 1 ホップ多い経路まで候補とした経路制御を行った場合に，リンク故障発生時に送信可能なトラヒックの合計を表す． これらの図より，いずれの場合も GFB222 が最も多くのトラヒックを送信可能であることがわかる．5.2 節で議論したように，GFB222 は，階層的な構造でスイッチ間のホップ数を小さくしつつ，各階層に複数のリンクを用いることで局所的な情報を用いた経路制御においても輻輳箇所の回避が容易なネットワーク構成となっている．そして，一部のリンクが故障

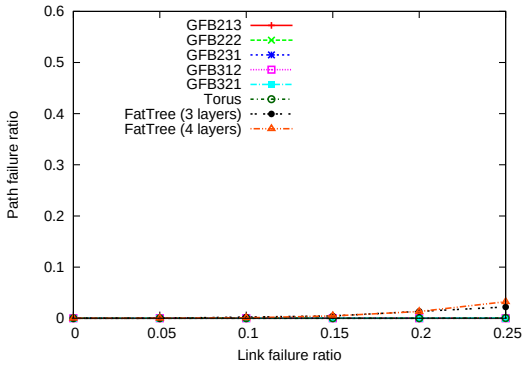
した場合においても，ホップ数や，各階層に構築されたリンク数は大きく変化しないため，GFB222 は，より多くのトラヒックを送信することができていると考えられる．



(a) 最短ホップ経路のみを考慮した場合

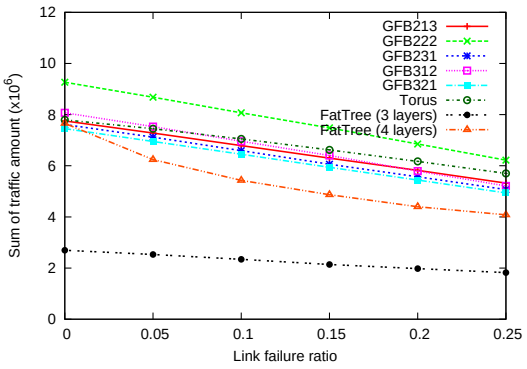


(b) 最短ホップ+1 ホップ以内の経路を考慮した場合

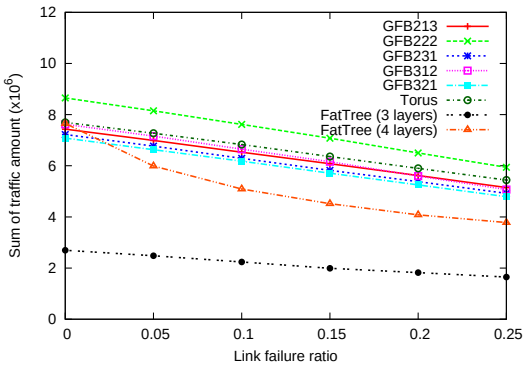


(c) 全ての経路を考慮した場合

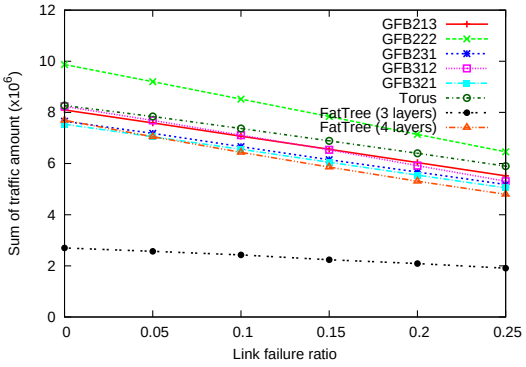
図 18: リンク故障が発生した場合の通信失敗率



(a) 局所制御

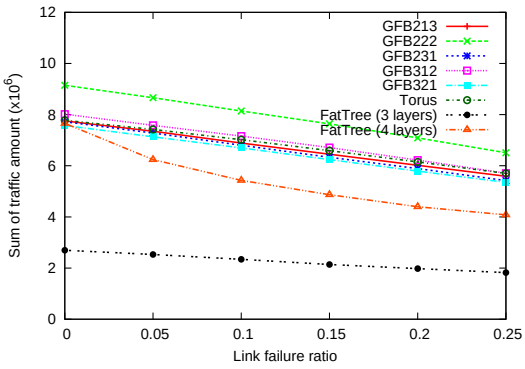


(b) ランダム制御

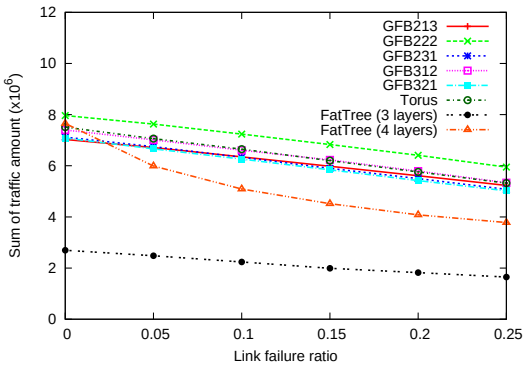


(c) 全体制御

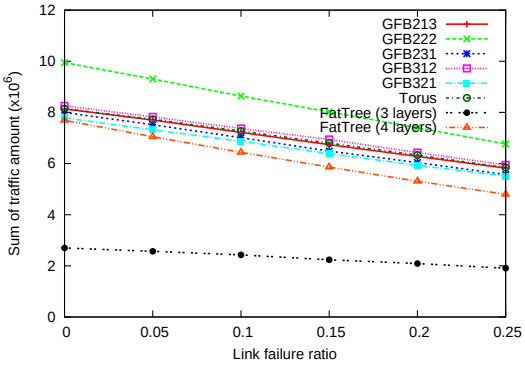
図 19: リンク故障が発生した場合の最短ホップ経路のみを転送先候補とした経路制御における各送信元・受信先間での送信可能なトラヒックの合計



(a) 局所制御



(b) ランダム制御



(c) 全体制御

図 20: リンク故障が発生した場合の最短ホップ+1 ホップ以内の経路を転送先候補とした経路制御における各送信元・受信先間での送信可能なトラヒックの合計

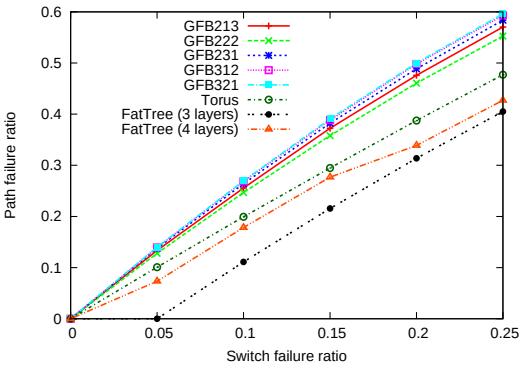
5.3.2 スイッチに故障が発生した場合

リンク故障の評価と同様，スイッチの故障が発生した場合についても，通信の失敗率を評価した．図 21(a) は最短ホップの経路のみ各スイッチが転送先の候補として保持している場合，図 21(b) は最短ホップ経路よりも 1 ホップ多い経路を転送先の候補として保持している場合，図 21(c) は全ての経路を転送先の候補としている場合の，スイッチ故障率に対する送信元・受信先間の通信の失敗率を示す．これらの図より，リンク故障の際と同様，最短ホップ経路のみを転送先候補として用いた経路制御では，GFB の通信失敗率は，FatTree や Torus よりも高くなっているものの，最短ホップ以外の経路も転送先候補として保持することにより，GFB においても通信失敗率を低下させることができることがわかる．そして，最短ホップよりも 1 ホップ多い経路も転送先候補として保持した場合，GFB の失敗率は Torus と同程度となることがわかる．

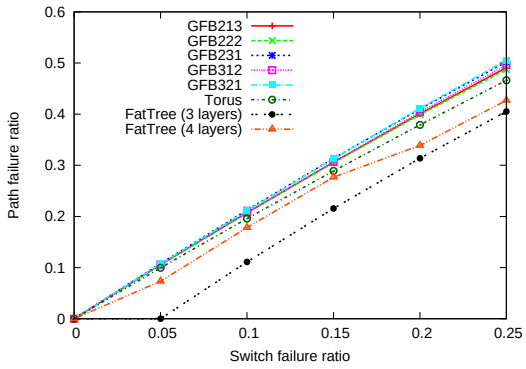
全ての経路を転送先の候補とした場合でも，GFB や Torus の失敗率は FatTree よりも高い．これは，GFB や Torus は全てのスイッチに直接サーバーが接続されているのに対して，FatTree ではサーバーが接続されているのは最下位層のスイッチのみであるためである．GFB や Torus では，1 台のスイッチが故障するとそのスイッチに接続しているサーバーとの通信は切断されてしまうが，FatTree で上位層のスイッチが故障しても，上位層の全てのスイッチが故障しない限り，通信が不可能なサーバーは生じない．その結果，FatTree の通信失敗率は小さくなる．しかしながら，FatTree では最下位層のスイッチのみしかサーバーに接続できないため，本評価のように GFB や Torus と同じ台数のスイッチを用いて接続した場合であっても，最下位層には，サーバー向けのポート数の大きな高機能なスイッチを用いる必要があり，設置コスト・消費電力ともに大きくなるという問題点がある．

スイッチが故障した場合に，送信可能なトラヒック量を比較した．図 22 は最短ホップ経路のみを，図 23 は最短ホップより 1 ホップ多い経路まで転送先候補として保持した場合に，スイッチ故障発生時に送信可能なトラヒックの合計を表す．また，図 24 は最短ホップ経路のみを，図 25 は最短ホップより 1 ホップ多い経路まで転送先候補として保持した場合に，スイッチ故障発生時の帯域圧迫トラヒックの最小値を表す．これらの図より，4 階層の FatTree が，送信可能トラヒックの合計の減少が最も少なく，帯域圧迫トラヒックの最小値は故障率が上がるとともに上昇していることがわかる．これは，FatTree では故障時にホップ数の大きいトラヒックの経路ほど切断されやすいためである．その結果，故障率が上がれば，ホップ数の大きいトラヒックの通信が減り，ホップ数の大きいトラヒックが帯域を消費していたリンクの帯域を他のホップ数の小さいトラヒックが利用可能となるため，送信可能なトラヒックが増加する．ただし，上述のように FatTree で GFB や Torus と同じ台数のサーバーを接続するためには，最下位層に高機能なスイッチを配置する必要があるという問題がある．

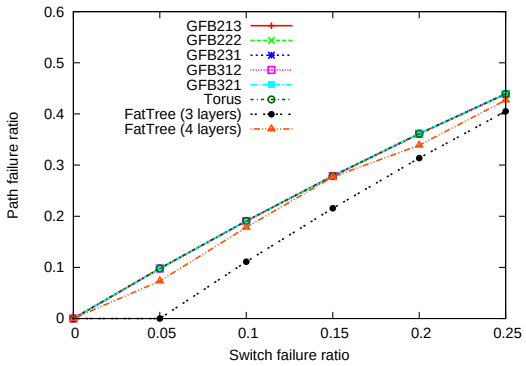
全スイッチにサーバーが接続している構造である，GFB や Torus で比較を行うと，故障が発生していない場合や，リンク故障の場合と同様，GFB222 が，送信可能なトラヒックの合計，帯域圧迫トラヒックの合計ともに最も大きいことがわかる．そのため，GFB222 のような，ホップ数が小さく，各階層に複数のリンクを持ち負荷分散が容易であるという構造は，スイッチ故障時にも維持することができると考えられる．



(a) 最短ホップ経路のみを考慮した場合

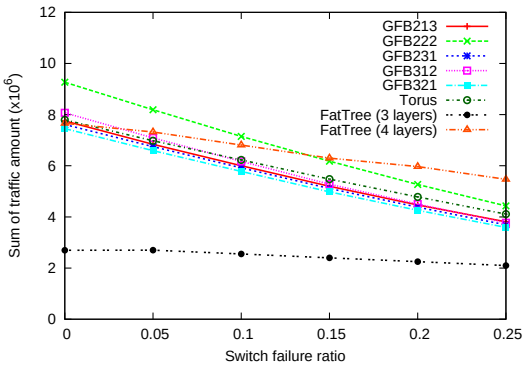


(b) 最短ホップ+1 ホップ以内の経路を考慮した場合

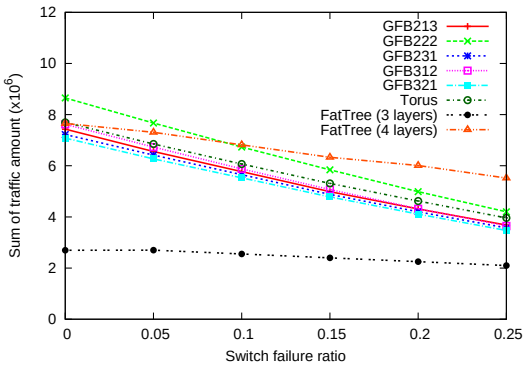


(c) 全ての経路を考慮した場合

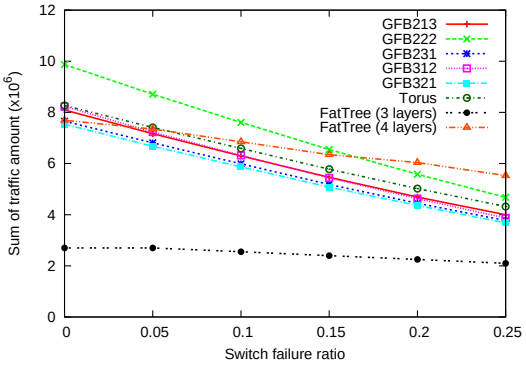
図 21: スイッチ故障発生時の通信失敗率



(a) 局所制御

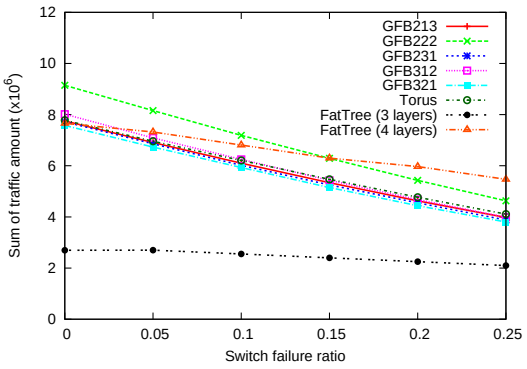


(b) ランダム制御

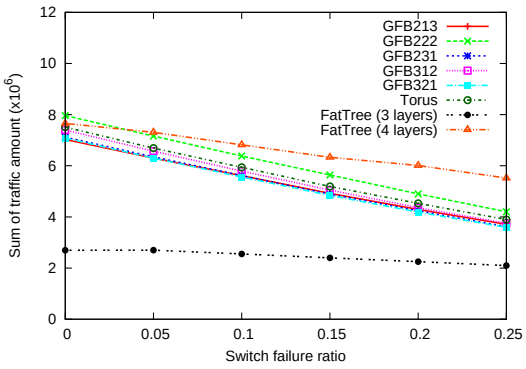


(c) 全体制御

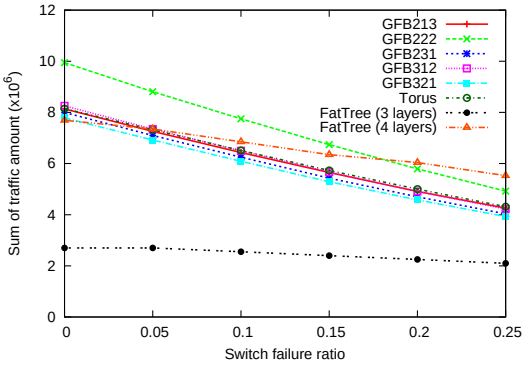
図 22: スイッチ故障が発生した場合の最短ホップ経路のみを転送先候補とした経路制御における各送信元・受信先間での送信可能なトラヒックの合計



(a) 局所制御

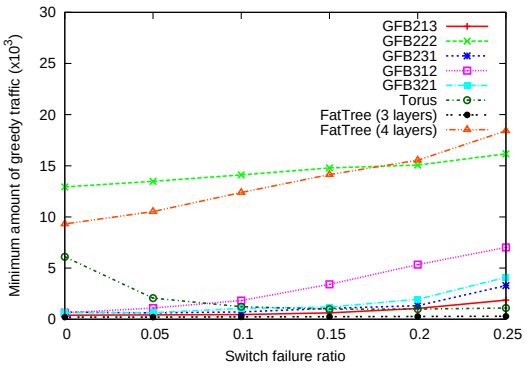


(b) ランダム制御

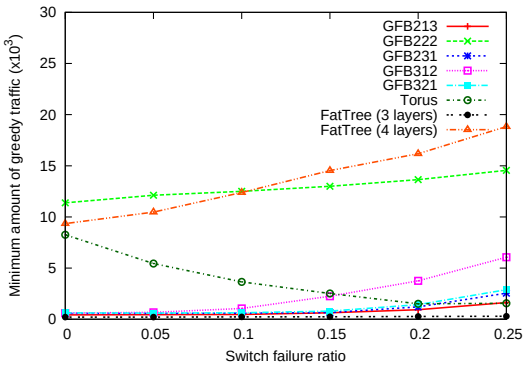


(c) 全体制御

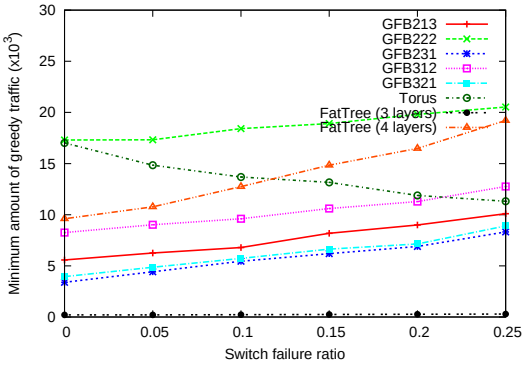
図 23: スイッチ故障が発生した場合の最短ホップ+1 ホップの以内の経路を転送先候補とした経路制御における各送信元・受信先間での送信可能なトラヒックの合計



(a) 局所制御

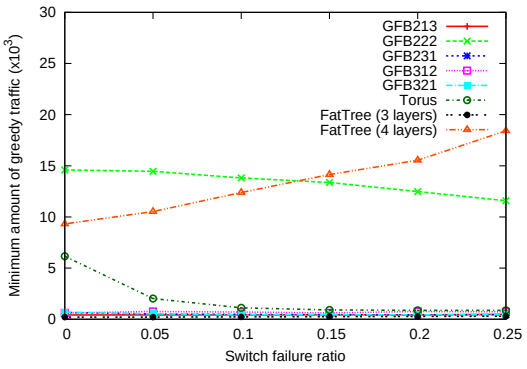


(b) ランダム制御

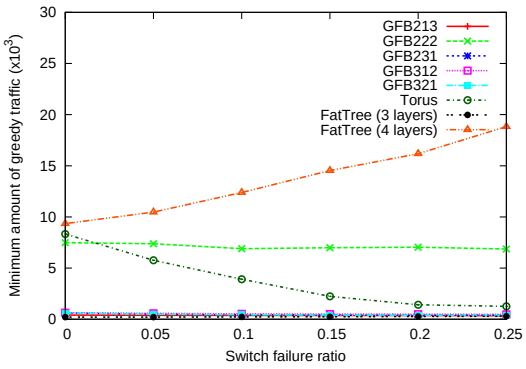


(c) 全体制御

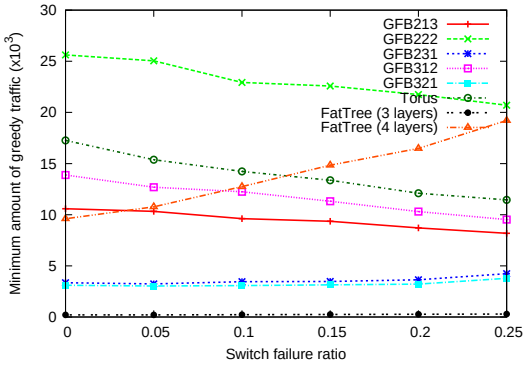
図 24: スイッチ故障が発生した場合の最短ホップ経路のみを転送先候補とした経路制御における帯域圧迫トラヒックの最小値



(a) 局所制御



(b) ランダム制御



(c) 全体制御

図 25: スイッチ故障が発生した場合の最短ホップ+1 ホップ以内の経路を転送先候補とした経路制御における帯域圧迫トラヒックの最小値

6　終わりに

本報告では，データセンターネットワークの構成と経路制御手法の組み合わせを評価した．本評価においては，FatTree と合わせて，GFB のパラメーターを変化させることにより生成した様々なネットワーク構成を評価対象のネットワーク構成として用いた．そして，各ネットワーク構成上で，集中制御により経路を決めた場合，トラヒック情報を用いずにランダムな負荷分散を行った場合，局所的なトラヒック情報を用いて経路を決めた場合に，サーバー間に確保可能な通信帯域やネットワーク内に収容可能な総トラヒック量の比較を行った．

評価の結果，集中型の全体制御は，最新のトラヒック情報を保持できる場合は，より多くのサーバー間のトラヒックを収容できるものの，トラヒック変動が発生した場合には，候補の転送先からランダムに転送先を選択する経路制御手法と同程度のトラヒックしか収容できないことが明らかとなり，環境変動の激しいデータセンターにおいては，リアルタイムに把握可能な局所的な情報を用いた経路制御が必須であることが明らかになった．

また，局所的な経路制御に適したネットワーク構成の評価を行った結果より，スイッチ間のホップ数を小さくすることが可能な GFB のような階層的に構築されたネットワーク構成が多くのトラヒックを収容するのに有効であることが明らかとなった．また，階層的に構築されたネットワークの中でも，各スイッチのポートのうち各階層の接続に用いられるポート数が均等な構成が，局所的な経路制御において，サーバー間により多くの通信帯域を確保できるということが明らかとなった．さらに，階層的に構築され，かつ，各スイッチのポートのうち各階層の接続に用いられるポート数が均等になるネットワーク構成はランダムなリンク故障やスイッチ故障が発生した場合にも，より多くのトラヒックを収容できる構造を維持できることが明らかになった．

さらに，局所情報を用いた経路制御では，いずれのネットワーク構成においても，最短ホップの経路のみではなく，最短ホップより 1 ホップ長い経路に含まれるスイッチも転送先スイッチの候補に含めることで，各サーバー間に確保可能な帯域の最小値の向上や，大きなトラヒックが発生しているスイッチの組み合わせが，全スイッチの組み合わせに占める割合が小さい場合に多くのリンクの帯域を有効に使うことによる収容トラヒック量の増加，さらに，故障発生時のサーバー間の接続性の維持に有効であることが明らかになった．

本報告では，データセンター内のサーバー間に確保可能な通信容量に基づき，経路制御手法とネットワーク構成の議論を行った．今後は，サーバー間の通信遅延も考慮した上で，低遅延の通信が可能なデータセンターネットワークを構築する際に必要な経路制御手法とネットワーク構成の組み合わせについて検討を行う予定である．

謝辞

本報告を終えるにあたりまして，日頃より熱心に指導して下さいました大阪大学大学院情報科学研究科の村田正幸教授，ならびに大阪大学大学院経済学研究科の大下裕一助教に厚く御礼申し上げます．また大阪大学大学院情報科学研究科の若宮直紀教授ならびに大阪大学大学院情報科学研究科の荒川伸一准教授には適切な助言を頂き，指導して頂きました．心より御礼申し上げます．さらに報告の作成にあたり，様々な助言を下さった樽谷優弥氏，吉成正泰氏をはじめとする村田研究室の方々にも御礼を申し上げます．

参考文献

[1] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM - 50th anniversary issue: 1958 - 2008*, vol. 51, pp. 107–114, Jan. 2008.

[2] J. Dean, "Designs, Lessons and Advice from Building Large Distributed Systems." `http://www.odbms.org/download/dean-keynote-ladis2009.pdf`.

[3] T. Benson, A. Anand, A. Akella, and M. Zhang, "The Case for Fine-Grained Traffic Engineering in Data Centers," in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, pp. 1–6, Apr. 2010.

[4] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The Nature of Datacenter Traffic: Measurement and Analysis," *ACM SIGCOMM Computer Communication Review*, pp. 202–208, Nov. 2009.

[5] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine Grained Traffic Engineering for Data," in *Proceedings of ACM CoNEXT*, pp. 1–12, Dec. 2011.

[6] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 63–74, Oct. 2008.

[7] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: A scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 75–86, Aug. 2008.

[8] Y. Liao, D. Yin, and L. Gao, "DPillar: Scalable Dual-Port Server Interconnection for Data Center Networks," in *Proceedings of 19th International Conference on Computer Communications and Networks*, pp. 1–6, Aug. 2010.

[9] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," in *Proceedings of the 34th annual international symposium on Computer architecture*, vol. 35, pp. 126–137, June 2007.

[10] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdats, "Hedera: Dynamic Flow Scheduling for Data Center Networks," in *Proceedings of the 7th*

USENIX conference on Networked systems design and implementation, pp. 281–295, Apr. 2010.

[11] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube:A high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 39, pp. 63–74, Aug. 2009.

[12] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, S. Lu, and J. Wu, "Scalable and cost-effective interconnection of data-center servers using dual server ports," *IEEE/ACM Transactions on Networking*, vol. 19, pp. 102–114, Feb. 2011.

[13] Y. Tarutani, "Virtual Network Topology Control to Achieve Low Energy Consumption in an Optical Data Center Network," Master's thesis, Osaka University, 2012.

[14] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," *ACM SIGCOMM Computer Communication Review*, vol. 39, pp. 51–62, Aug. 2009.

[15] The BlueGene/L Team IBM and Lawrence Livermore National Laboratory, "An Overview of the BlueGene/L Supercomputer," in *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pp. 1–22, Nov. 2002.

[16] M. Yokokawa, F. Shoji, A. Uno, M. Kurokawa, and T. Watanabe, "The K computer: Japanese next-generation supercomputer development project," in *Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*, pp. 371–372, Aug. 2011.

[17] S. Young and S. Yalamanchili, "Adaptive routing in generalized hypercube architectures ," in *Proceedings of the Third IEEE Symposium on Parallel and Distributed Processing*, pp. 564–571, Dec. 1991.

[18] C. HOPPS, "Analysis of an Equal-Cost Multi-Path Algorithm," RFC 2992, Internet Engineering Task Force, Nov. 2000. `http://tools.ietf.org/html/rfc2992`.

[19] A. Singh, *LOAD-BALANCED ROUTING IN INTERCONNECTION NETWORKS*. PhD thesis, Stanford University, 2005.