

リード・ソロモン符号化複数経路マルチキャスト転送の 最適スケジュールを生成するためのブロック割当法

倉田 真之^{1,a)} 柴田 将弘^{1,b)} 鶴 正人^{1,c)}

概要: 単一送信者から多数の受信者へ同一ファイルを配布する一対多ファイル転送において、各受信者のファイル取得完了時間の理論最小値の実現を目指し、各受信者への Max-Flow 量から成る複数経路を同時利用する複数経路マルチキャスト転送 (MPMC) を検討する。先行研究では、リード・ソロモン符号化 MPMC (Coded-MPMC) とそのスケジュール決定のためのブロック割当の受信者順序を動的に変更するヒューリスティクスを提案し、OpenFlow に基づく実装を試作した。しかし、大規模で複雑なネットワークの中には、これらの手法だけでは最適スケジュールを生成できないものが存在した。本報告では、既存手法の失敗例の分析から得られた新しいブロック割当法を導入することで、最適スケジュールが見つけれれることを、多数のネットワークトポロジに対してシミュレーションを行い検証する。

Block allocation for Multipath-Multicast transfer with Reed-Solomon Coding to generate an optimal schedule

Abstract: In one-to-many file transfer where a single sender delivers a file to a number of recipients, a Multipath-Multicast transfer (MPMC) scheme is considered in order to realize each recipient's theoretical minimum retrieval completion time using the max-flow paths from the sender. We proposed the MPMC with Reed-Solomon coding (Coded-MPMC) and its scheduling with some heuristics in changing the block allocation order dynamically. However, the previously proposed scheduling does not find an optimal schedule on some large-scale complicated network topologies. Therefore, in this report, we improve the coded-MPMC scheduling by introducing a new block allocation method, and verify that it can find an optimal schedule on many network topologies based on real-world networks through simulation.

1. はじめに

クラウド、分散コンピューティングおよびコンテンツ配信ネットワーク (CDN) 技術の普及に伴い、分散配置されたデータセンタやデータセンタ内のサーバ間での、データやソフトウェアの共有、複製そして移動の必要性が高まっている。また、IoT エッジクラウドコンピューティング技術では、地理的に離れた場所に分散した多数の受信者に対して、大規模な配信をリアルタイムに行う必要がある。そのため、データセンタ内だけではなくデータセンタ間でも、単一の送信者がネットワーク経路を介して、大容量のファイルを複数の受信者に配信する、高速・高効率な一対多ファイル

転送の重要性が高まっている。

一対多ファイル転送では、信頼性が保証されたマルチキャスト転送が基本的に用いられる。しかし、単一のマルチキャスト木を使用する場合 [1]、全受信者の取得完了時間は各受信者のネットワーク的位置空間に依存し、全体の取得完了時間が最悪値となる受信者によって決定される。この欠点に対処するために、各受信者のネットワーク的位置空間に基づき、適切にグルーピングされた複数のマルチキャスト木の使用が検討された [2]。

グローバルに分散されたデータセンタ間での一対多ファイル転送の需要が増加しているため、SDN (Software-Defined Networking) 技術の 1 つである OpenFlow を活用した、データセンタ間専用の高速バックボーンネットワークが展開されている [3]。SDN 技術は柔軟で適応性のあるルーティングを実現可能にするため、複数のマルチキャスト木を利用する様々な手法が検討されている。例えば [4] では、各受信

¹ 九州工業大学院 情報工学府 先端情報工学専攻
680-4 Kawazu, Iizuka, Fukuoka, Japan

a) kuratam@infonet.cse.kyutech.ac.jp

b) shibata@cse.kyutech.ac.jp

c) tsuru@cse.kyutech.ac.jp

者のネットワーク的位置空間とトラフィックの分布に従って複数のマルチキャスト木を使用することで、ネットワーク上で連続した複数の一対多ファイル転送を効率的且つ迅速に実行できる。ただし、これらの手法は送信者から各受信者へのネットワーク経路の総容量である Max-Flow(MF)量を十分に利用していないため、各受信者の取得完了時間を最小化していない。

そこで、私たちは全二重有線リンクで完全に制御された OpenFlow ネットワーク上で、単一の送信者から複数の受信者にファイルを配信するための、複数経路マルチキャスト (MPMC: Multipath-Multicast) 一対多ファイル転送モデルを提案した [5]。MPMC では、転送ファイルは均等長のブロックに分割され、一つまたは複数のフェーズで各受信者に転送される。各フェーズで複数のブロックを MF 経路で同時転送すると共に、複数の受信者へ同一ブロックをマルチキャスト転送することで、無駄な重複転送を防ぐことができる。

スケジューラである OpenFlow コントローラ (OFC) はネットワークトポロジ、リンク容量、送信者と受信者の位置をスケジューリング前に把握しており、フェーズ毎にどのブロックをどの経路を利用して転送するかスケジューリングを決定する。このプロセスはブロック割当 (BA: Block Allocation) と呼ばれ、全受信者が全フェーズで MF 経路を完全に利用する BA を行い、各受信者の取得完了時間が下限値となる最適スケジューリングを生成することを目的とする。

しかし、[6] の単純な MPMC では、各受信者の前フェーズまでに取得したブロックの違いによって、続くフェーズで MF 経路を完全には利用できない BA を行ってしまう場合がある。そこで、私たちはこの問題を解決するために、始点ネットワーク符号化を導入したリード・ソロモン符号化 MPMC(Coded-MPMC)[6] を提案した。Coded-MPMC では、送信者は事前にオリジナルブロックからリード・ソロモン符号化を用いて必要な数の符号化ブロックを生成し、各フェーズで転送する。そして、受信者はオリジナルブロック個数分の異なるブロックを受け取ることで、元のファイルが復元可能となる。この性質を利用することで、受信者間の取得ブロックの違いを考慮せず BA が行え、最適スケジューリングの生成が容易となる。

また [6] では、大規模な実世界のネットワークトポロジに対して、Coded-MPMC で最適スケジューリングを生成するために、BA を行う受信者順序を必要に応じて静的/動的に変更するヒューリスティクスを検討した。しかし、この Coded-MPMC は MF 経路をスケジューリング開始時に求めており、高次数のノードを多く含むより大規模で複雑な実世界のネットワークトポロジに対して、受信者順序の変更を何度実行しても、最適スケジューリングが生成できない場合が存在した。

本報告では、これらの問題を解決するために、既存の BA

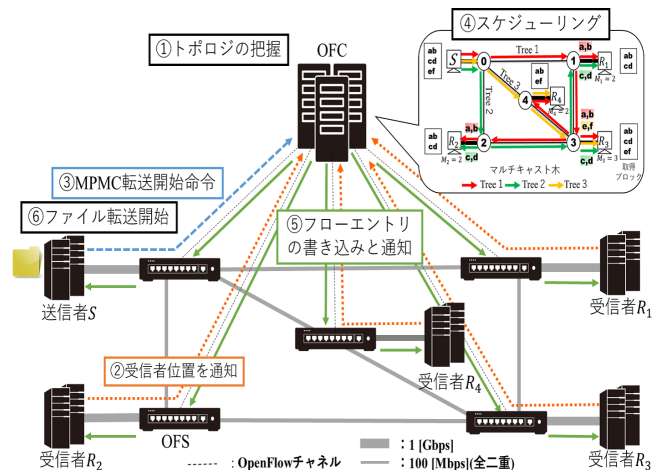


図 1 OpenFlow を用いた MPMC ファイル転送モデルの全体像

法の失敗例の分析から得られた新しい BA 法を提案する。その手法では、MF 経路をスケジューリング開始時に作成するのではなく、各受信者の BA を行う直前に、BA に使用するリンクやノードといったリソースの消費を抑えた MF 経路を作成し、MF 経路を完全に利用する BA を行えない受信者に対して経路変更を行う。そして BA を行った後、その受信者の BA が他の受信者に対する、最適スケジューリングとなるための制約に違反するかを逐次検査し回避する。

以下、2 節で MPMC ファイル転送と従来の BA 法を説明する。そして、3 節で提案する新しい BA 法について述べ、4 節で提案する BA 法を導入した Coded-MPMC の有用性を確かめるために、*The Internet Topology Zoo*[8]にあるデータセット内の多数のネットワークトポロジに対してシミュレーション評価を行う。最後に 5 節でまとめを行う。

2. MPMC ファイル転送

2.1 MPMC ファイル転送モデル

表 1 に定義する変数を用いて、MPMC ファイル転送の説明を行う。MPMC では、単一の送信者 S が N_R 人の受信者 $R_1, R_2, \dots, R_{N_R}$ に対して、全二重リンクを持つ OpenFlow ネットワーク上でファイルを転送する。このとき、 S や R_i と OpenFlow スイッチ (OFS) 間のリンクは、ボトルネックにならないほど十分に大きい容量を持っている。また、モデルを簡潔化するためにパケットロスが発生しないネットワークを想定する。

図 1 は、MPMC ファイル転送モデルの全体像を表している。まず、OpenFlow コントローラ (OFC) は、ネットワークトポロジ・受信者の位置を把握する。そして、送信者から転送開始命令を受け取るとスケジューリングを開始し、各 OFS へのフローエントリの書き込みと各ホストへの通知を行う。通知を受け取った受信者は受信準備を行い、送信者はファイル転送を開始する。

MPMC では、ファイルの転送時に全受信者の最大のリンク容量を利用するために、 S から R_i への Max-Flow(MF)

表 1 変数の定義 Part.1

変数	定義
S	送信者ホスト
R_i	各受信者ホスト ($i \in \{1, 2, \dots, N_R\}$)
L	S が転送するファイルのサイズ
b_i	S から R_i への Max-Flow(MF) 経路の総パス容量
B	全ての b_i の最大公約数
M_i	R_i の Max-Flow(MF) 量 (B に対する b_i の比)
K	全受信者の MF 量の最小公倍数 (ブロック数)
T_i	R_i の最良転送完了時間
K_i	R_i の未取得ブロック数

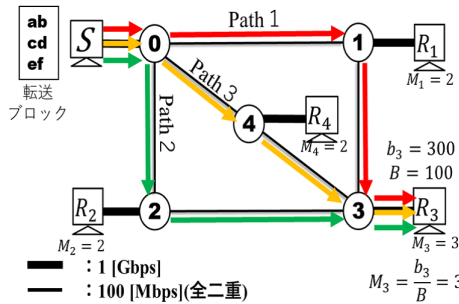


図 2 S から R_3 への MF 経路

経路を利用してファイルが転送される。このとき各経路に沿った最小のリンク容量をパス容量、全 MF 経路を用いて転送する場合の総同時転送容量を総パス容量と呼ぶ。受信者が一人の場合は、Edmonds-Karp のアルゴリズム [7] を用いることで、容易に MF 経路が導出される。しかし、受信者が複数存在する場合は、同様に導出することが難しいため、次小節で紹介する貪欲なアルゴリズムを利用する。

MPMC における重要な性能評価指標は、 R_i の取得完了時間であり、それは S がファイルを転送し始めてから、 R_i がファイル全体を取得するまでの時間と定義される。 R_i の取得完了時間の下限値は、 R_i の総パス容量を完全に使用してファイル全体を取得する時間で、 L/b_i と表される。また、 S が転送するファイルは、全受信者の M_i の最小公倍数である K 個に分割され、各ブロックサイズは L/K となる。

図 2 は S から R_3 への MF 経路を示している。このとき、 B, b_3, M_3 はそれぞれ 100[Mbps], 300[Mbps], 3 となる。また、 R_1, R_2, R_4 の b_i, M_i はそれぞれ 200[Mbps], 2 となるため、ファイルは $K(=6)$ 個のブロックに分割される。

2.2 MPMC ファイル転送スケジューリング

MPMC では、均等長のブロックを複数の受信者に対して、一つまたは複数フェーズを用いて転送する。そして、各フェーズで送信者は複数のブロックを受信者に MF 経路を用いて転送すると同時に、他受信者の MF 経路を効率的に利用するために、マルチキャスト転送を行う。

図 1 で示すように OpenFlow コントローラ (OFC) は、送信者がブロックを転送する前に、各フェーズでどのブロックをどの MF 経路で転送するかをスケジューリングする。

このプロセスを、ブロック割当 (BA : Block Allocation) と呼び、全受信者の全ブロックの取得が完了されるまで実行される。以下に、MPMC のスケジューリング方法を示す。

- (1) BA を行う受信者順序を決定するために、 K 個のブロックを取得していない受信者 R_i の最良転送完了時間 T_i を導出する。 T_i は未取得ブロックを R_i の MF 経路を完全に利用して取得する時間で、 $T_i = \left(\lceil \frac{K_i}{M_i} \rceil \times \frac{L}{BK} \right)$ と定義される。この T_i の昇順が BA を行う受信者順序 (BA 順序) となる。
- (2) BA が最初に行われる受信者は主受信者と呼ばれ、ブロック割当は主受信者とそのフェーズでファイルの取得を完了するように設計される。つまり、主受信者が全未取得ブロックの取得を完了するまでの時間が、各フェーズの期間となる。また同値の T_i を持つ受信者が複数存在する場合は、事前受信者優先度と呼ばれるスケジューリングの入力として与える、ランダムな受信者順序を基に決定される。
- (3) 残りの未受信者らはその BA 順序に従って、受信者へブロックを届けることができる未割当容量が正のリンクがそのフェーズに存在する限り、MF 経路の一つに (部分的に) 重なる可能性がある BA 済み経路上の未取得ブロックが割り当てられる。これにより、主受信者の MF 経路は残りの未受信者らによって分岐し、複数のマルチキャスト木が形成される。
- (4) (1)~(3) の未受信者に対する貪欲な BA は、全受信者がファイル全体を取得するまで繰り返される。

全受信者の取得完了時間が下限値となるスケジュールが生成された場合、そのスケジュールは最適スケジュールとなる。また、あるフェーズの受信者に対する BA で、全 MF 経路を完全に利用するブロック割当は BE(Bandwidth Efficient)-BA と呼ばれ、全受信者に対して BE-BA を行える場合は GBE(Globally BE)-BA と呼ばれる。全フェーズで GBE-BA を実現することは、全受信者が全フェーズで MF 経路を完全に利用してブロックを得られることを意味し、最適スケジュールを生成するための必要十分条件である。

2.3 Coded-MPMC ファイル転送

単純な MPMC ファイル転送では、前フェーズの未取得ブロックの違いにより GBE-BA を見つけられず、最適スケジュールを生成できないネットワークポロジが存在する。そこで最適スケジュールを効率的に生成するために、リード・ソロモン符号化複数経路マルチキャスト (Coded-MPMC) が提案された [6]。

Coded-MPMC では、送信者はリード・ソロモン符号化を利用して、 K 個の全てのオリジナルブロックから必要な数の符号化ブロックを転送前に生成する。全受信者がファ

イルの取得を完了するまで、1 フェーズ目はオリジナルブロックが、続くフェーズでオリジナルブロック、もしくは符号化ブロックが転送される。各受信者はオリジナルブロック、もしくは符号化ブロックを合計 K 個取得することにより、元のファイルを復元可能となる。この性質を用いることで、各受信者の前フェーズの未取得ブロックの違いを考慮することなく、BA を実行できる。これにより、MPMC における BA の制限が緩和され、最適スケジュールを効率的に見つけることができる。さらにその緩和に加えて、符号化ブロックを用いることで、前フェーズで行った BA を続くフェーズでも再現可能となる。そのため、Coded-MPMC では1 フェーズ目に GBE-BA となる複数のマルチキャスト木を発見することができれば、続くフェーズも1 フェーズ目と同様のマルチキャスト木を利用することで GBE-BA を再現できる。つまり、Coded-MPMC で最適スケジュールを生成するためには、1 フェーズ目に GBE-BA を見つけることが重要となる。

各フェーズで転送されるブロックは、作成されたマルチキャスト木と、そのマルチキャスト木から取得する未受信者の未取得ブロックに依存する。OFC は各木が転送する未受信者の未取得ブロックを調べ、各木に前フェーズまでに転送したオリジナルブロック、もしくは符号化ブロックが転送可能かを調査する。そして転送可能な場合はそれらのブロックを、一方で不可能な場合は、新たな符号化ブロックを用いる。この符号化ブロック生成アルゴリズムを利用することで、徒に符号化ブロックを生成することを避けられ、送信者は最低限の符号化ブロック数で転送が可能となる。そのため、スケジューリング後に各ホストが、実際に符号化ブロックをエンコード・デコードする時間が短縮される。

2.4 Coded-MPMC での BA 例

図2のネットワークポロジで、事前受信者優先度 $[R_1, R_2, R_3, R_4]$ が入力された時の、Coded-MPMC の BA 例を示す。まず、 R_i の K_i と M_i から、 T_i を導出する。このとき $T_3 < T_1 = T_2 = T_4$ であるため、与えられた事前受信者優先度から BA 順序は (R_3, R_1, R_2, R_4) となる。そして、決定された BA 順序に従って、送信者 S は主受信者 R_3 の MF 経路を用いてブロックをマルチパス転送すると共に、残りの未受信者らにマルチキャストでブロックを転送する。これにより、図3に示す3つのマルチキャスト木「Tree1」「Tree2」「Tree3」が形成され、 S は R_3 に全ブロックを転送するために、 $\lceil \frac{K_3}{M_3} \rceil = 6/3 = 2$ 個のブロックを各木で転送する。また、このフェーズの BA は全受信者に対して BE-BA が行っているため、GBE-BA であるといえる。

前フェーズの BA が GBE-BA であるため、続く2 フェーズ目では未受信者らに対して、各受信者が1 フェーズ目に取得したブロックとは異なるブロックを得られるように同

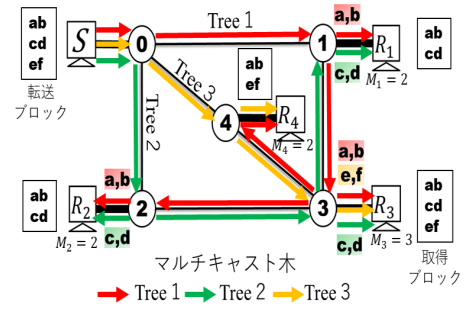


図3 Coded-MPMC での BA(1 フェーズ目)

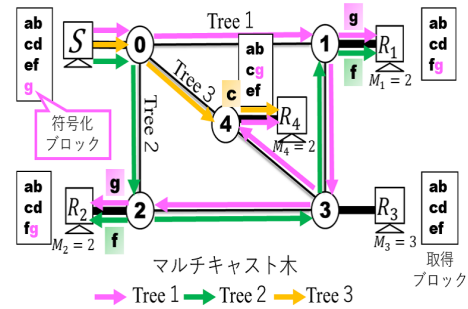


図4 Coded-MPMC での BA(2 フェーズ目)

様の BA を行う (図4)。マルチキャスト木「Tree2」「Tree3」の場合は、各木がブロックを転送する未受信者らに対して、転送済みのブロックが転送可能であるので、それぞれ“f”、“c”を転送する。一方で「Tree1」の場合は、その木から取得する全未受信者が、新たに取得可能な転送済みのブロックは存在しないので、新たに符号化ブロック“g”を用いて転送する。これにより全フェーズで GBE-BA が実現され、最適スケジュールの生成が容易となる。

2.5 BA の受信者順序の変更

各フェーズにおける BA 順序は、スケジューリング開始時に R_i の T_i と事前受信者優先度から決定される。そしてその BA 順序に従って、別の受信者に対する BA が BE-BA であるかを考慮せずに、後戻りすることなく BA が実行される。しかし、大規模で複雑なネットワークポロジの場合、同値の T_i を持つ受信者が複数存在する。そのため事前受信者優先度次第で、様々な複数のマルチキャスト木が形成され、効率的な最適スケジュールの生成が難しい。そこで、BA 順序を静的に管理する「Path-length ascending order allocation」と、動的に管理する「Deprioritized reallocation」「Prioritized reallocation」の3つのヒューリスティクスを提案した [6]。

「Path-length ascending order allocation」では、BA を MF 経路が通過する各リンクで消費するリンク容量の総和の小さい受信者から行うために、MF 経路の平均長の昇順を第一のキー、最大長の昇順を第二のキーとして事前受信者優先度をソートする。これは、最適スケジュールもしくはそれに近いスケジュールを調査した中で、MF 経路の平均長・最大長の昇順に BA が行われているものが多いとい

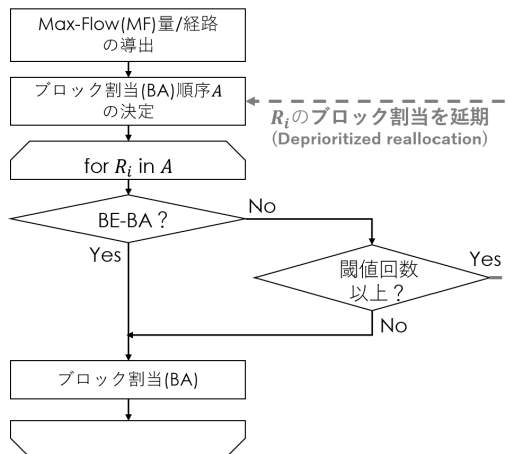


図5 BAのフローチャート(既存手法)

う観察に基づいて検討された。

他二つの方法は、BA順序の動的変更に着目したものである。BE-BAを行えない受信者が存在するとき、そのままBAを実行するのではなく、BA順序の動的変更を行うことで、スケジュールの探索範囲を拡張することを目的とする。「Deprioritized reallocation」では、BE-BAを行えない受信者 R_i を、同値の T_i を持つBA順序の末尾に置く。これにより、 R_i はBAが延期され R_{i+N} ($N > 0$) となる。そして R_{i+N} は、それ以前の受信者のBAから形成されたマルチキャスト木から未取得ブロックを転送されることで、BE-BAが行われる可能性がある。この際、BAの延期が何度も実行されるのを避けるために閾値を設けている。この手法を導入した時の、BAのフローチャートを図5に示す。

また「Prioritized reallocation」は、全受信者に対して「Deprioritized reallocation」が適用されてもなお、BE-BAとはならない受信者が存在する場合に適用される。この手法では、BE-BAを行えなかった受信者 R_i を、同じ T_i を持つBA順序リストの先頭に置いて同一フェーズを始めからもう一度行う。これにより、 R_i は R_i 以前にBAが行われた受信者の影響を受けずに済み、BE-BAを行える可能性がある。この手法は、不必要なループを避けるために一度のみ実行される。

3. 提案するBA法

前節で紹介したヒューリスティクスを導入したCoded-MPMCでは、単純なMPMCで最適スケジュールを生成できなかった大規模で複雑なネットワークポロジに対して、効率的に最適スケジュールを生成することが可能になった。しかし、この手法では、*The Internet Topology Zoo*[8]にあるデータセット内の高次数のノードが多く存在する、より複雑なネットワークポロジに対して、1フェーズ目にGBE-BAを実現できずに最適スケジュールが生成されない場合が存在した。そこで、GBE-BAを実現できないシナリオを複数分析し、既存のBA法の課題を抽出した。

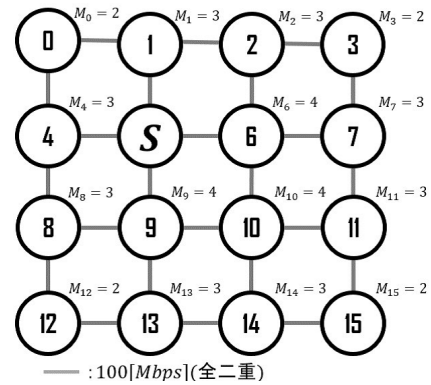


図6 4×4の格子状ネットワーク

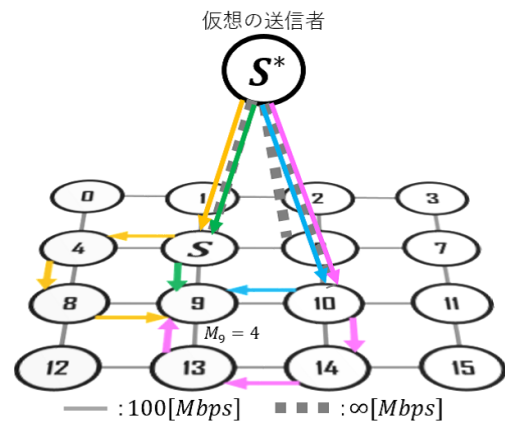


図7 S^* から R_9 へのMF経路 (R_{10} と R_6 のBA後)

抽出された課題とその改善手法の説明を、図6の格子状ネットワークを用いて行う。ノード⑤は送信者 S の直結ノード、それ以外のノード①は受信者 R_i の直結ノードを表している。このとき送信者は、MF量2,3,4を持つ受信者が存在するので、ブロックを $K(=12)$ 個のブロック $(a, b, \dots, l)$ に分割して転送を行う。

3.1 MF経路の作成

図5で示すように、既存手法ではスケジュールリング開始時に導出されたMF経路を用いて、BE-BAとはならない受信者のBAを行う順番を動的に変更することでGBE-BAを探索していた。しかし、そのようなMF経路ではBA順序の変更を何度行ったとしても、GBE-BAを実現できない場合が存在した。その主な原因として、各受信者のBAで消費する各リンクのリンク容量の総和が余剰であることが挙げられる。そのため、後にBAが行われる受信者が自身の総パス容量を十分に使用することができずに、BE-BAとはならない場合が存在する。

そこで提案手法では、必要最小の総リンク容量でBE-BAを実現するために、各受信者がBAを行う直前にMF経路を導出する。その手法では、 S とそのフェーズで K 個のブロックを既に取得している受信者の直結ノードを、仮想の送信者 S^* とコスト0、リンク容量 ∞ の仮想リンクで結ぶ。 S^* を設けることで、単一の送信者 S から作成するMF経

路では使用されなかったノードから、各受信者が未取得ブロックを最短経路で取得することが期待される。これにより、各受信者が BA で使用する総リンク容量を抑えられるため、後に続く未受信者らが BA に使用するノードとリンクの候補を増やすことができ、各受信者の BE-BA の実現が容易となる。

仮想の送信者 S^* を用いたときの、 R_9 への MF 経路を図 7 に示す。このとき、 R_6, R_{10} は既に BA が行われ、全ブロックを取得している。そのため、 S^* をそれらの受信者の直結ノードと、リンク容量 ∞ の帯域で結ぶ。

3.2 最大 2 部マッチング問題を利用した BA

既存手法では R_i に対して BA を行う際、各経路で消費する総リンク容量を考慮していない。そこで消費する総リンク容量を減らすために、重み付き最大二部マッチング問題を利用した BA 法を提案する。使用する二部グラフは、作成された MF 経路とブロックの 2 つの頂点集合から形成され、各経路が転送可能な R_i の未取得ブロックに対して有向辺が結ばれる。そして各有向辺には、消費するリンク数の総和に応じて重みが付けられており、最小コストでマッチング数が最大化されるペアを見つけることを目的とする。得られたペア数が、 R_i の未取得ブロック数と一致する場合は、 R_i に対して BE-BA を行える。一方で一致しない場合は、「Deprioritized reallocation」を用いて R_i の BA を延期する。

3.3 MF 経路の変更

ある受信者の BA の延期が閾値回数以上行われた時、これ以上 BA の延期を繰り返しても BE-BA は実現できないと判断し、MF 経路の変更を試みる。

R_i への BA に使用可能な (未割当容量が正である) MF 経路の内、未取得ブロックを転送できない MF 経路が 1 つ (最大 2 部マッチング問題を解いたとき、 R_i の最大マッチング数 -1) の場合は、その経路の置換を試みる。まず、 R_i までの BA が終了したネットワークに、 R_i の BA で使用する各リンクのリンク容量を消費した状態のネットワークを作成する。次に、 R_i の未取得ブロックを所持しているノードを、仮想の送信者 S^* とコスト 0/リンク容量 ∞ の仮想リンクで結ぶ。最後に、 S^* から R_i への未取得ブロックを転送可能な新たな経路を作成する。これにより、未取得ブロックを転送可能な MF 経路に置換でき、 R_i に対して BE-BA を行える。

一方で、未取得ブロックを転送できない使用可能な MF 経路が 2 つ以上ある場合、それらの経路が経由するリンクに対して、十分に大きいコストを設定する。重み付きのネットワークを利用することで、BE-BA とはならない MF 経路を再び作成されることを避けることが可能となり、BE-BA となる MF 経路が新たに作成されることが期待される。

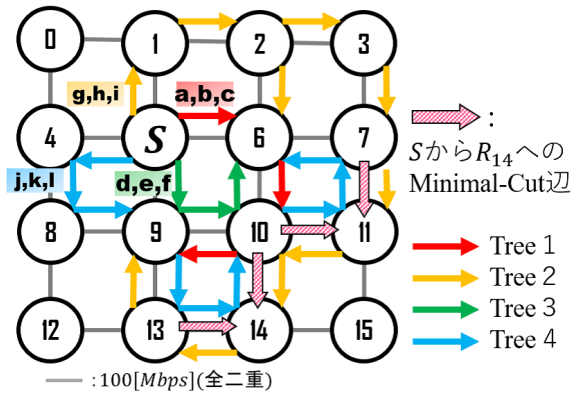


図 8 R_9 に対する BA 後のネットワーク

表 2 変数の定義 Part.2

変数	定義
MC_i	R_i の任意の Minimal-Cut 辺集合
AMT_i	MC_i の中で、既に割当てられているマルチキャスト木の集合
RMC_i	MC_i の中で、空きのリンク容量がある辺の集合
PT_i	R_i が既に所持しているブロックを転送したマルチキャスト木の集合
E_{PT_i}	PT_i が経由する辺の集合
G	入力として与えるグラフ (ネットワークトポロジ)
G_i	R_i までの BA が終了した状態のグラフ
$M_i(graph)$	入力された $graph$ 上での R_i の MF 量

3.4 GBE-BA となるための条件

図 8 は R_{10}, R_6, R_9 の順で BA を行った後の、ネットワークの状態を表している。 R_9 は経路 $[10 \rightarrow 9]$ を用いてブロック (a, b, c) を、経路 $[10 \rightarrow 14 \rightarrow 13 \rightarrow 9]$ を用いてブロック (g, h, i) を取得する。このとき、 R_{14} はブロック (g, h, i, j, k, l) を既に取得しているが、 11 から (a, b, c) もしくは (d, e, f) を取得できず、MF 経路を完全に利用できない。そのため、 R_{14} に対して BE-BA を行えない。

R_{14} に対して BE-BA が行われない理由を、表 2 の変数を用いて述べる。図 8 の S から R_{14} への Minimal-Cut (MC) 辺集合 MC_{14} に着目した時、 R_{14} に対して BE-BA を行うためには、 $M_{14}(= 3)$ 種類のマルチキャスト木が割当られる必要がある。しかし、 MC_{14} に対して 2 種類しかマルチキャスト木が割当てられていない。つまり BE-BA を行うためには、 R_i の任意の MC_i に M_i 種類以上のマルチキャスト木を割当て、もしくは割当る余地を残す必要がある。この R_i が BE-BA となるための制約は式 (1) で表され、GBE-BA を実現するためには全受信者の MC_i は式 (1) を満たす必要がある。

$$|AMT_i| + |RMC_i| \geq M_i \quad (1)$$

図 8 で示すように、受信者 R_i への BA がより後で BA を行う受信者 R_j に対するこの制約に違反し、必ず GBE-BA が行われない場合が存在する。本来は GBE-BA を実現す

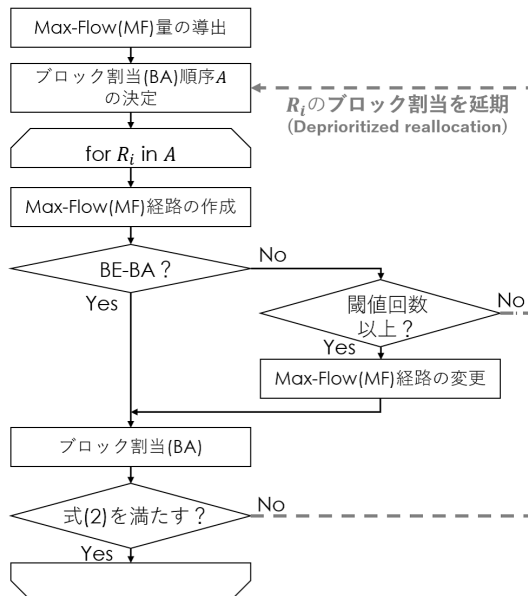


図9 BAのフローチャート(提案手法)

るために、全受信者が式(1)を満たす経路作成を行う、もしくはBA後に式(1)を満たさない全ての MC_i を探し出しBAのやり直しを行いたい。しかし、全受信者の MC_i を導出した後、ある受信者のBA毎に全ての MC_i を調査してBAを行うのは、問題となっている MC_i を見つけることが困難である且つ、膨大な時間を要するため非現実的である。

そこで、式(2)を用いて R_i までのBAが終了した状態のグラフ(ネットワーク) G_i 上で、残りの未受信者 R_j に対してBE-BAを行えるかどうかを、式(2)を用いて判定する。式(2)の左辺は G_i から EP_{T_i} を消去したグラフ上での R_j のMF量、右辺は R_j のMF量と PT_j の種類数の差を表しており、 R_i のBAが終了した後、 R_j が必要数の未取得ブロックを得るためのマルチキャスト木が分岐可能かを調査する。式(1)とは異なり全ての MC_j を調査することなく、 R_j に対してBE-BAが実現可能かを検知できる。

$$M_j(G_i - EP_{T_i}) \geq M_j(G) - |PT_j| \quad (2)$$

式(2)を満たさない R_j がいる場合は、 R_i のBAを取り消しそのBAを延期する。そして新たに R_i のMF経路を作成する際は、再び同じ経路を作成しないために、 R_i に対するBAで使用可能であった経路が経由する各リンクに対して、十分に大きいコストを設定する。これにより、受信者 R_j が式(2)を満たすMF経路を作成することが期待され、GBE-BAとならないことを回避することができる。

最後に、この節で紹介した全手法を導入した、提案するBA法を図9に示す。

4. 性能評価

4.1 評価方法

提案するBA法(図9)を導入したCoded-MPMC一対

表3 調査したデータセット

ノード数	データセット名 ([8])
40~49 個	Bell Canada, China Telecom, Lambdanet, PalmettoNet, UUNET, GENAT, Renater, RoEduNet, SWITCH
50~59 個	IRIS Networks, Bell South, BTN, Tinet, DFN, Hibernia Atlantic, CESNET, CUDI, SURFNET
60~69 個	FORTHNET, Internode, ASNET-AM, ESnet, GARR, Uninett, Missouri Network Alliance
70~79 個	Columbus Networks, Intellifiber, Sinet Syringa Networks, Tecove (level3)
80~99 個	Bestel, Viatel, ULAKNET
100 個以上	Interoute(100 個), ITC Deltacom(113 個), ION(120 個), TATA(145 個), GTS CE(149 個), Colt Telecom(153 個), US Carrier(158 個), Cogent(197 個)

多ファイル転送の有用性を調査するために、多数の実世界のネットワークポロジが格納されている *The Internet Topology Zoo*[8] のデータセットを使用し、GBE-BAが見つけれられるかを検証する。既存研究では“Renater”、“SWITCH”、“Columbus Networks”の3種のみ扱っていたが、本報告ではその3種を含めた42種のデータセットを使用する。それらは、*The Internet Topology Zoo*中の非連結ではない、ノード数が40~200個の全てのデータセットである(表3)。また、各リンクのリンク容量が異なるシナリオでもGBE-BAが見つけれられるかを調査するために、データセットに定義されている辺の重複数に応じて、リンク容量を100/200/300[Mbps], 100/400/1000[Mbps]に変更する。

スケジューリングの入力として与える事前受信者優先度は、[6]で効率的にGBE-BAを探索することが可能であると検証された「Path-length ascending order」で決定する。また、送受信ホストは各ノードにボトルネックにならないほど十分に大きいリンク容量で接続されており、単一の送信者は送信者の直結ノードを除く、全てのノードに接続されている受信者に一対多ファイル転送を行う。

4.2 評価結果

提案するBA法を導入したCoded-MPMCを用いることで、全てのトポロジの任意の送信者位置で1フェーズ目にGBE-BAを発見できることを確認した。それらの例として、データセット“Tinet”、“Interoute”で送信者位置(ノードID)を9,54としたときの、GBE-BAを図10, 11に示す。

そして表4に、送信者が1000[Mbytes]のファイルを転送する際に分割するオリジナルブロック数と生成する符号化ブロック数を、単純なマルチキャストと最適スケジュールを用いたCoded-MPMCで転送する場合の全受信者の平均取得完了時間を示す。これらの結果から、提案するBA法を導入したCoded-MPMCを用いることで、単純なマルチキャスト転送より素早く転送を行えることが分かる。

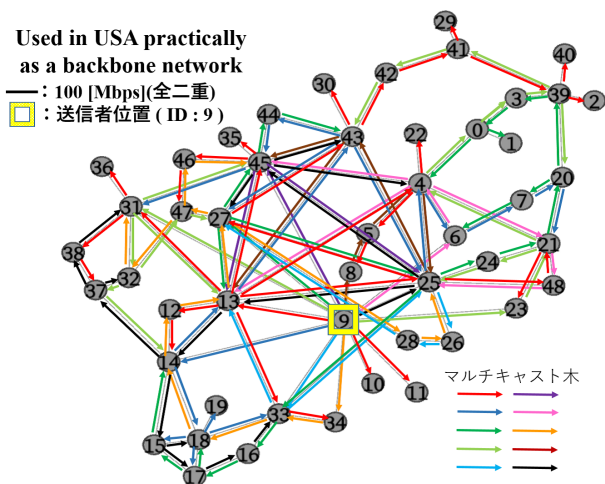


図 10 Tinet(送信者位置:ID 9) での GBE-BA

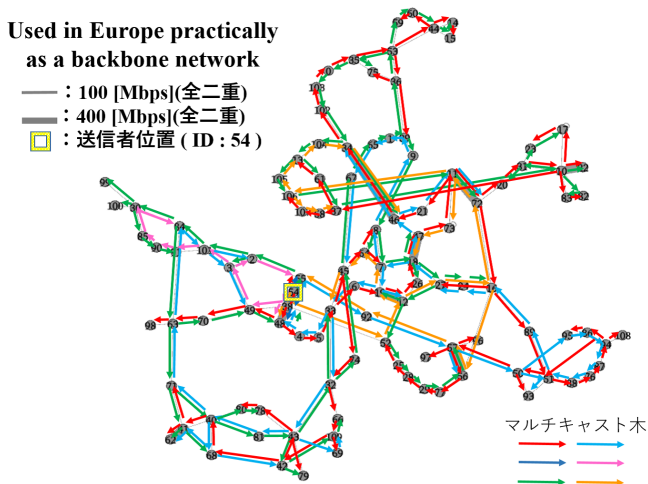


図 11 Interroute(送信者位置: ID 54) での GBE-BA

加えて、提案する BA 法を用いることによって、全シナリオ (約 4000 種類) で BA 順序の動的変更を行うヒューリスティクスの 1 つである「Prioritized reallocation」を実行せずに、GBE-BA が達成できることを確認した。これにより、同一フェーズが再び実行されなくなり、スケジューリング時間の減少につながる。しかし、各受信者の BA 毎に全未受信者に対して式 (2) を用いた検証を行うため、全体的にみると大幅に計算時間が増加する。その対策としては、オンラインではなくオフラインであらかじめスケジューリングを実行しておくことが挙げられる。また調査したところ、式 (2) を満たさないシナリオは、わずか 12 種類しか存在しない。そのため、基本的には式 (2) を用いた検証を行わずに、オンラインでスケジューリングを実行可能であると考えられる。

5. まとめ

本報告では、Coded-MPMC 一対多ファイル転送で最適スケジュールを生成するために、1 フェーズ目に GBE-BA を発見することを目的として、既存のブロック割当 (BA) 法での失敗例から課題を抽出し、新たな BA 法を提案した。

表 4 シミュレーションシナリオと実行結果

データセット	Tinet	Interroute
送信者位置	ID 9	ID 54
送信ファイルサイズ	1000[Mbytes]	
オリジナルブロック数	1260[個]	12[個]
符号化ブロック数	771[個]	12[個]
単純な Multicast での 平均取得完了時間	80.0000[s]	80.0000[s]
Coded-MPMC での 平均取得完了時間	40.6892[s] (下限値)	35.9633[s] (下限値)

その手法では、各受信者の BA に使用するノード・リンクといったリソースの消費を抑え、各受信者に対する BA が GBE-BA となる条件を満たすかどうかの検証を行う。そして、実際に提案手法を導入した Coded-MPMC を用いることで、多数の実世界のネットワークトポロジに対して、1 フェーズ目に GBE-BA を見つけられるというシミュレーション結果を示した。

今後は、提案する BA 法の有用性をより確かなものにするために、Barabasi-Albert モデルといった著名なランダムネットワークでの検証を考えている。また OpenFlow を用いて実装した MPMC, Coded-MPMC の実現可能性を調査するために、大規模な SDN テストベッドである “RISE” [9] 上で検証を行う。

参考文献

- [1] K. Miller, K. Robertson, et al., “StarBurst Multicast File Transfer Protocol (MFTP) Specification,” Internet Draft, draft-miller-mftp-spec-03.txt, April 1998.
- [2] S. Bhattacharyya, J. Kurose, et al., “Efficient rate-controlled bulk data transfer using multiple multicast groups,” IEEE/ACM Trans. Networking 11(6):895–907, 2003.
- [3] S. Jain, A. Kumar, et al., “B4: Experience with a Globally-deployed Software Defined WAN,” Proc. ACM SIGCOMM’13, pp.3–14, 2013.
- [4] M. Noormohammadpour, C. S. Raghavendra, et al., “QuickCast: Fast and Efficient Inter-Datacenter Transfers using Forwarding Tree Cohorts,” Proc. IEEE info-com’18, pp.225–233, 2018.
- [5] A. Nagata, Y. Tsukiji, M. Tsuru, “Delivering A File by Multipath-Multicast on OpenFlow networks,” Proc. IEEE INCoS13, pp.835–840, 2013.
- [6] M. Kurata, K. Heira, M. Shibata, M. Tsuru “Minimizing One-to-many File Transfer Times using Multipath-Multicast with Reed-Solomon Coding,” Proc. the 31th International Teletraffic Congress (ITC-31), 2pages,
- [7] J. Edmonds, R. Karp “Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems,” Proc. Journal of the ACM, pp. 248–264, 1972.
- [8] “The Internet Topology Zoo,” [Online]. Available: <http://www.topology-zoo.org/>
- [9] “RISE ポータルサイト,” [Online]. Available: <https://www.jgn.nict.go.jp/rise/index.html>