一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
ICM2021-17(2021-07)

送信者符号化複数経路マルチキャストに基づく多対多ファイル転送

伊藤　幸輝[†]　柴田　将拡[†]　鶴　　正人[†]

† 九州工業大学情報工学府 〒820-8502 福岡県飯塚市川津 680-4
E-mail: †ito.koki233@mail.kyutech.jp, ††{shibata,tsuru}@cse.kyutech.ac.jp

あらまし　データセンタ内や地理的に分散配置されたデータセンタ間での大容量ファイルの共有，複製または移動のためのトラヒックが急激に増加しているため，そのようなデータ転送に掛かる時間を短縮することが喫緊の課題となっている．我々の研究グループの先行研究では，帯域が保証された全二重リンクからなる SDN 上での単一の送信者から複数の受信者へのスケジュールされたファイル転送（一対多ファイル転送）を検討し，送信者符号化複数経路マルチキャスト転送（Coded-MPMC) 手法を開発した．Coded-MPMC では，送信者から各受信者への Max-Flow 量を使い切るような転送が行われることで，各受信者が理論最小時間でファイル取得を完了することが数多くのトポロジにおいて検証された．しかし，実際のネットワークでは単一の一対多ファイル転送だけではなく，同時に複数の一対多ファイル転送の要求も発生する．そこで本報告では，Coded-MPMC を基づき，高速で高効率な複数の一対多ファイル転送（多対多ファイル転送）をスケジューリングするための，送信者符号化多対多転送（Coded Many-to-Many Transfer; C-M2MT）手法を提案する．基本的な設計を紹介し，C-M2MT を逐次一対多転送と比較した基礎検討を示す．
キーワード　マルチキャスト転送，複数経路転送，多対多ファイル転送，一対多ファイル転送，Max-Flow 問題，送信者符号化

Many-to-many file transfers based on multipath multicast with sender coding

Koki ITO[†], Masahiro SHIBATA[†], and Masato TSURU[†]

† Computer Science and Systems Engineering, Kyushu Institute of Technology 680–4 Kawazu, Iizuka–shi, Fukuoka, 820–8502 Japan
E-mail: †ito.koki233@mail.kyutech.jp, ††{shibata,tsuru}@cse.kyutech.ac.jp

Abstract　In response to a rapid growth of the traffic demand for duplicating, migrating, or sharing large-sized files among multiple servers in a datacenter and across geographically distributed datacenters, it is a big challenge to reduce the time taken in such bulk data transfers. In our previous work, we consider a scheduled transmission of a file from a single sender to multiple recipients (one-to-many file transfer) in Software-defined networks (SDNs) with bandwidth-guaranteed full-duplex links, and developed Coded Multipath Multicast (Coded-MPMC) scheme. In Coded-MPMC, each recipient can fully utilize the Max-Flow value of transmission from the sender and thus can achieve a lower-bound of its file retrieval completion time, which was verified to a large number of topologies. However, in reality, multiple one-to-many file transfers co-exist simultaneously on a network. In this report, therefore, we propose a scheduling scheme, Coded Many-to-Many Transfer (C-M2MT), based on Coded-MPMC, for a fast and efficient transmission of files from multiple senders to multiple recipients (many-to-many file transfer). A basic design of C-M2MT and its preliminary evaluation compared with a sequentially applied one-to-many file transfers are provided.
Key words　Multipath transfer, Multicast transfer, Many-to-many file transfer, One-to-many file transfer, Max-Flow problem, Sender coding

1. はじめに

データセンタ内や地理的に分散配置されたデータセンタ間での，大容量ファイルやソフトウェアの共有，複製，または移動によるトラヒック量の急激な増加に対応するために，高速かつ高効率な一対多ファイル転送の重要性が高まっている [1]．我々の研究グループでは，全二重リンクからなる SDN 上での単一の送信者から複数の受信者へのスケジュールされたファイル転

送（一対多ファイル転送）を検討してきた [2]．特に，先行研究 [3] では全二重ネットワーク上で，各受信者が送信者からの Max-Flow 量（以下，MF 量）を完全に利用して，理論最小時間でファイル取得を完了するために，送信者符号化一対多転送（以下，Coded-MPMC）による一対多転送が提案され，数多くのトポロジにおいて最適スケジュールが生成可能であることが示されている．しかし，実際のネットワークでは単一の一対多ファイル転送だけではなく，同時に複数の一対多ファイル転送の要求が想定される．そこで，別の先行研究では，送信者符号化を用いない複数経路マルチキャスト転送 (MPMC) に基づき多対多のファイル転送のスケジューリングを検討した [4]．それに対し，本報告では，Coded-MPMC に基づく送信者符号化多対多転送（以下，C-M2MT）を提案する．C-M2MT の基本的な設計を紹介し，C-M2MT と各送信者からの Coded-MPMC 転送を送信者数分繰り返す符号化逐次 1 対多転送（以下，C-SSMT）手法をシミュレーションによって比較し，基礎検討を示す．

2. 送信者符号化一対多転送

送信者は転送ファイルを均等長に分割した N 個のオリジナルブロックから，Reed-Solomon（RS）符号を用いて符号化ブロックを生成する．その後 1 フェーズもしくは複数フェーズを使用して，複数種類のブロックのマルチパス転送と同一ブロックのマルチキャスト転送によりリンク容量を効率的に利用したブロック転送を行い，各受信者がそれらのマルチキャストフローから N 個のオリジナルブロックもしくは符号化ブロックを受信するまでブロック転送を行う．最後に，N 個のブロックを受信した受信者は，RS 符号化を用いることで受信ブロックからファイルを復元する．最適なブロック転送スケジュールの場合，各フェーズの未受信者はそれぞれの MF 量を満たすマルチキャストフローから異なるブロックを受信し，下限値（転送ファイルサイズと MF 量の商）でファイルの取得を完了する．

3. 送信者符号化多対多転送

3.1 手法の説明

C-M2MT は次の手順で行われる．なお，対象とするトポロジやリンク容量，受信者の位置は全て既知であり，トポロジ中のリンクは全て帯域が保証された全二重リンクとする．

（1） 送信者優先度の決定

まず n 人分のファイル転送要求に対して，ブロック割当を行う順番を決めるために，送信者 $S_j (j = 0, \ldots, n-1)$ に優先度をつける必要がある．優先度の決定方法は次の 2 つのどちらかを用い，優先度が最も高い送信者を主送信者と呼ぶ．

(A) 最大 MF 量（送信者 S_j から各受信者 $R_k (k = 0, \ldots, n-1)$ への MF 量 $M_{j,k}$ の最大値）の大きい順

(B) 予想最小転送時間 T_j（送信者 S_j の持つファイル長 B_j と S_j から各受信者 R_k への MF 量 $M_{j,k}$ の最大値の商）の小さい順

（2） ファイル分割数の決定

最初の主送信者 S_j のファイルは $M_{j,k}$ の最小公倍数 LCM の個数分のオリジナルブロックに分割する．主送信者 S_j の転送ファイル長を LCM で割った値をブロック長とし，主送信者以外の送信者 (副送信者）のファイルは，ファイル長をブロック長で割った値（割り切れない場合は商 +1）の個数分のオリジナルブロックに分割する．

（3） 符号化ブロックの生成

符号化が必要な送信者はオリジナルブロックから符号化ブロックを生成する．

（4） ブロック割当

ある主送信者 S_j からの $M_{j,k}$ が最大の受信者 R_k を主受信者と呼び，MF 量を満たす Max-Flow 経路を使って全ブロック（符号化ブロックを含む場合は任意の異なるオリジナルブロック個数分）を主受信者に転送するようにスケジューリングする．この時に主受信者以外の受信者（副受信者と呼ぶ）へもマルチキャスト経路を使って可能な限り転送のスケジュールを決定する（これらの転送を合わせてメイン転送と呼ぶ）．メイン転送のスケジューリング終了後，未使用のリンク容量があれば，送信者優先順に副送信者からの転送（サブ転送と呼ぶ）を行うためのスケジュールを決定する．主受信者が主送信者からの全ブロックを受信し終えるまでの時間区間をフェーズと呼ぶ．そのフェーズでの主受信者が主送信者の全ブロックを受け取ると主受信者を変更して次のフェーズが始まり，全受信者が全ブロックを受け取ると主送信者を変更して次フェーズが始まる．これを全受信者が全送信者のファイルを受け取るまで繰り返す．

C-M2MT 手法の一連の流れを図 1 に示す．

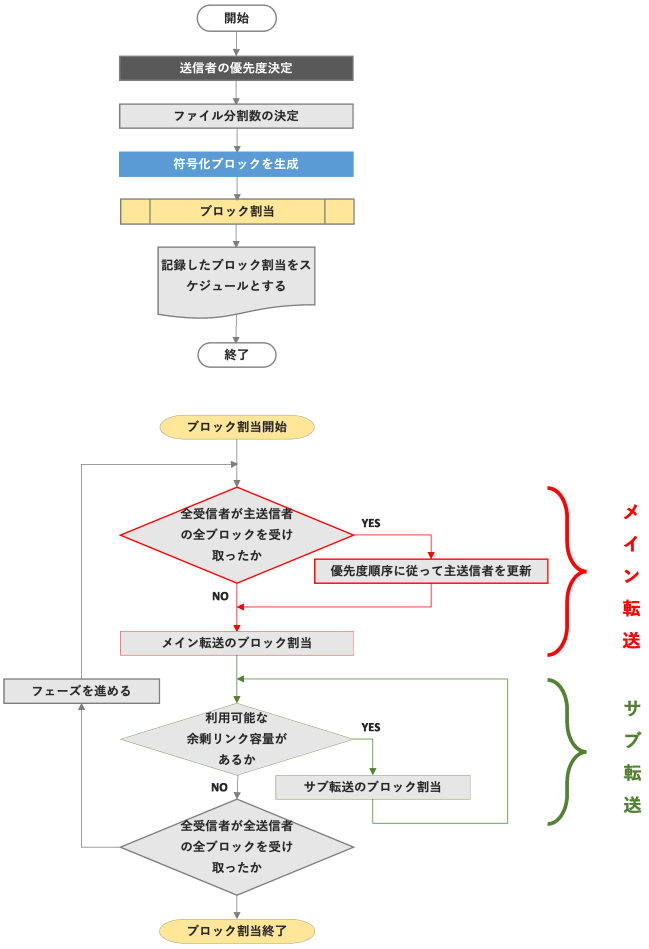

図 1　C-M2MT 手法のフローチャート

3.2　送信者 5 人が異なるファイルを持ったトポロジでの例

図 2 のトポロジを例に C-M2MT 手法について具体的に説明する．なお，この例での送信者優先度の決定方法は（A）最大 MF 量の大きい順とする．5 台のホストとスイッチ（0,1,2,3,4）間が 1[Gbps]，スイッチ間が 100[Mbps] で全二重接続され，送信者 S_0,S_1,S_2,S_3,S_4 がそれぞれ異なるファイル（転送ファイル長は S_0 が 120[MB]，S_1 が 40[MB]，S_2 が 40[MB]，S_3 が 120[MB]，S_4 が 40[MB]）を自身以外の受信者（送信者 S_0 の場合は R_1,R_2,R_3,R_4）へ転送を行う．各送信者 $S_j (j = 0,\ldots,4)$ から受信者 $R_k (k = 0,\ldots,4)$ への MF 量 $M_{j,k}$ を計算すると，送信者の優先度は $S_0 > S_3 > S_1 > S_2 > S_4$ となる．優先度の一番高い送信者 S_0 が主送信者となり，主送信者の持っているファイルは $\{M_{0,k} | k = 1,\ldots,4\}$ の最小公倍数である 6 個のオリジナルブロック（$\alpha_{0,\ldots,5}$）に分割される．さらに，ファイルの分割数は主送信者 S_0 の 1 ブロック長である 120[MB] / 6 = 20[MB] に合わせて他の送信者のファイルも分割すると，各ブロック数は S_1 が 2（$\beta_{0,1}$），S_2 が 2（$\gamma_{0,1}$），S_3 が 6（$\delta_{0,\ldots,5}$），S_4 が 2（$\varepsilon_{0,1}$）となる．次に，符号化が必要な送信 S_0 と S_3 はオリジナルブロックから符号化ブロック（S_0 は α_6，S_3 は δ_6）を生成する（図 2）．

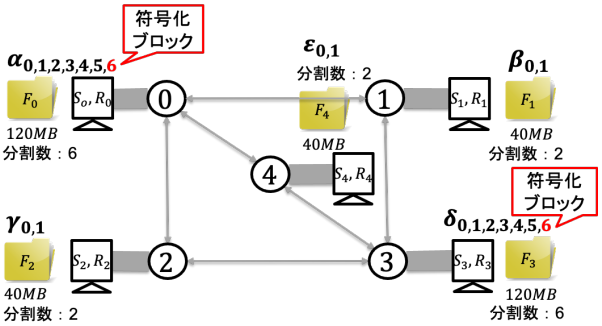

図 2　C-M2MT 手法のファイル分割と符号化ブロック生成

1 フェーズ目では，主送信者 S_0 から図 3 のようなメイン転送（主受信者は R_3）によるブロック割り当てと，副送信者 S_1，S_2，S_4 から図 4 のようなサブ転送によるブロック割り当てが同時に行われる結果として 1 フェーズ目終了時には，R_0 は F_1 の $\beta_{0,1}$（F_1 受信完了），F_2 の $\gamma_{0,1}$（F_2 受信完了），F_4 の $\varepsilon_{0,1}$（F_4 受信完了）を，R_1 と R_2 は F_0 の $\alpha_{0,1,2,3}$ を，R_3 は F_0 の $\alpha_{0,1,2,3,4,5}$（F_0 受信完了）を，R_4 は F_0 の $\alpha_{0,1,4,5}$ を持っていることになる．

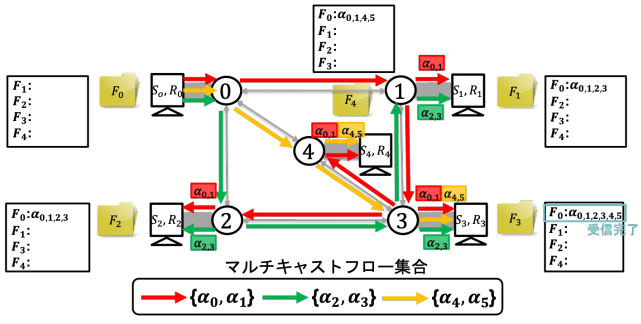

図 3　C-M2MT の 1 フェーズ目（メイン転送）

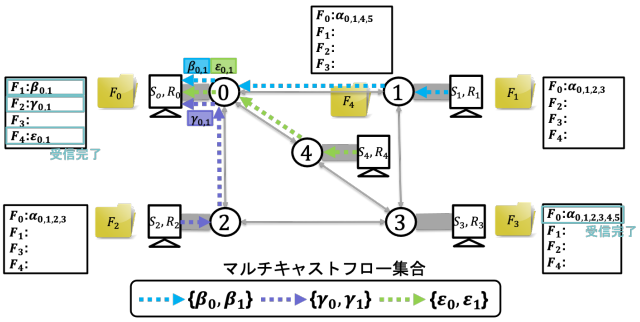

図 4　C-M2MT の 1 フェーズ目（サブ転送）

2 フェーズ目では，主送信者 S_0 からメイン転送（主受信者は R_1）と，副送信者 S_4 からサブ転送により，図 5 のようなマルチキャストフローでブロックが割り当てられる．結果として 2 フェーズ目終了時には，R_0 は F_1 の $\beta_{0,1}$，F_2 の $\gamma_{0,1}$，F_4 の $\varepsilon_{0,1}$ を，R_1 と R_2 は符号化ブロック α_6 を含む F_0 の $\alpha_{0,1,2,3,4,6}$（F_0 受信完了）を，R_3 は F_0 の $\alpha_{0,1,2,3,4,5}$（F_0 受信完了），F_4 の ε_0 を，R_4 は，符号化ブロック α_6 を含む F_0 の $\alpha_{0,1,2,4,5,6}$（F_0 受信完了）を受け取っていることになる．

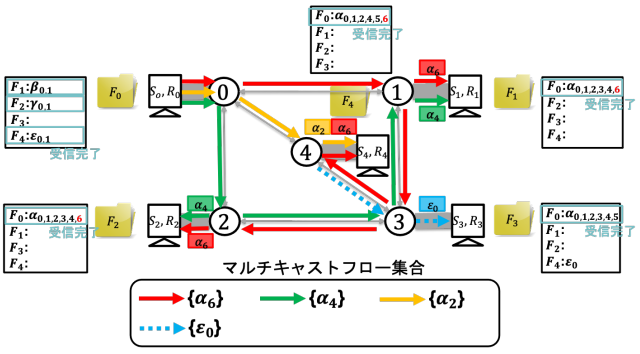

図 5　C-M2MT の 2 フェーズ目（メイン + サブ転送）

3 フェーズ目では，S_0 からの全ブロックを全ての受信者 R_1，R_2，R_3，R_4 が受け取ったため，送信者優先度に従い，S_3 を送信者に変更し，1，2 フェーズ目と同様にブロック割当を行う．

このようにして全送信者の全ブロックを全ての受信者が受け取るまで続け，最終的には 6 フェーズで終了する．ここまでの転送スケジューリングのタイムチャートを図 6 に示す．

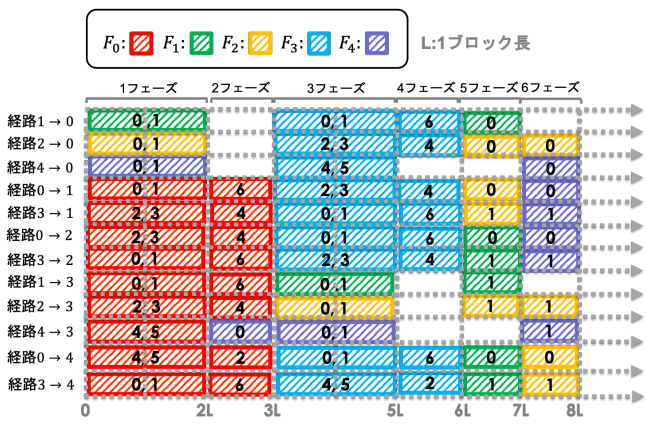

図 6　C-M2MT のスケジューリングのタイムチャート（A 方式）

4. シミュレーションによる評価

4.1 シミュレーション方法

次の 3 つ（図 7，図 8，図 9）のトポロジ上でシミュレーションを行う．なお，これらの図のキャプションは (全ノード数)-(送信者数) を表す．

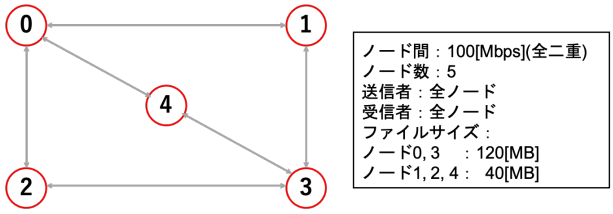

図 7　トポロジ 5-5

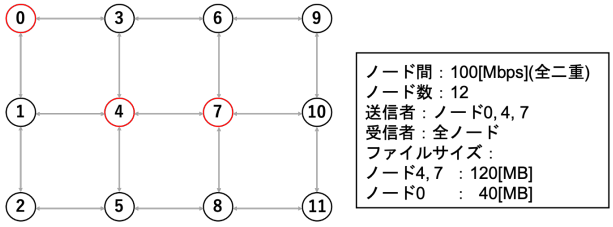

図 8　トポロジ 12-3

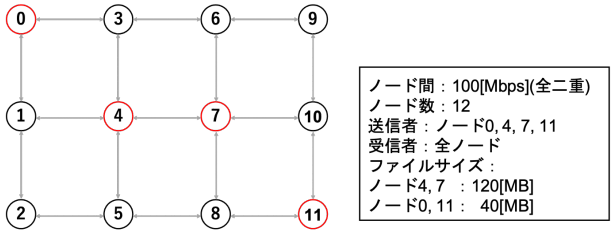

図 9　トポロジ 12-4

その際，次の 2 つの転送手法を用いる．なお，本稿で提案する C-M2MT 手法を以降「提案手法」と呼ぶ．一方，それと比較するために，各送信者からの Coded-MPMC 転送を送信者数分繰り返す C-SSMT 手法を「単純手法」と呼ぶ．

- 単純手法：C-SSMT
- 提案手法：C-M2MT

送信者優先度の決定方法は次の 2 つを用いる．

- A 方式：各送信者の最大 MF 量の大きい順
- B 方式：各送信者の予想最小転送時間の小さい順

シミュレーションでの性能指標は次の 3 つとする．

- 各送信者の全転送完了時間：
  各送信者から見て全ての受信者へファイル転送が完了するまでの時間
- 各送信者の平均転送完了時間：
  各送信者から見て各受信者へファイル転送が完了するまでの時間の平均
- 各送信者の転送ブロック数：
  各送信者から見て全ての受信者へファイル転送が完了するまでに送った合計のブロック数

4.2 シミュレーション結果

トポロジ 5-5（図 7）でのシミュレーションの結果を表 1，表 2，表 3 に示す．

表 1　トポロジ 5-5 での各送信者の全転送完了時間

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S0	4.80	4.80	S1	1.60	1.60
S3	9.60	9.60	S2	3.20	3.20
S1	11.20	11.20	S4	4.80	3.20
S2	12.80	12.80	S0	9.60	8.00
S4	14.40	12.80	S3	14.40	12.80

表 2　トポロジ 5-5 での各送信者の平均転送完了時間

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S0	4.40	4.40	S1	1.60	1.60
S3	9.20	9.20	S2	3.20	3.20
S1	11.20	8.40	S4	4.80	2.80
S2	12.80	8.80	S0	9.20	7.60
S4	14.40	9.20	S3	14.00	12.40

表 3　トポロジ 5-5 での各送信者の転送ブロック数

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S0	8	8	S1	2	2
S3	8	8	S2	2	2
S1	2	6	S4	2	4
S2	2	6	S0	8	8
S4	2	6	S3	8	8

表 1 より，単純手法と比較して提案手法の方が最終的な転送完了時間が小さくなっていることが分かる．これは，提案手法のサブ転送が有効的に機能したからであると考えられる．送信者優先度の決定方法である A 方式と B 方式で比較すると，最終的な転送完了時間は同じだが，1 番目から 4 番目の送信者の全転送完了時間は B 方式の方が小さいことが分かる．これは B 方式で決定される送信者の順序が予想転送完了時間の小さい順であるためである．

また，表 2 より，A 方式と B 方式ともに 3 番目の送信者から先は，単純手法と比較して提案手法の方が平均転送完了時間が小さくなっていることが分かる．これは 1，2 番目の送信者が主送信者であったフェーズのサブ転送で，それ以降の送信者のブロックのいくつか（または全て）が先に転送されたからであると考えられる．

しかし，平均転送完了時間が小さくなった 3 番目の送信者か

ら先は，表 3 より，各送信者の転送ブロック数は単純手法よりも，提案手法の方が多くなっている．これは，提案手法でサブ転送が発生し，単純手法よりもブロックを転送する機会が増加したからである．よって，ファイルの転送完了時間と転送ブロック数はトレードオフの関係であると考えられる．また，転送ブロック数について A 方式と B 方式で比較すると，A 方式の方が多くなっている．これは，A 方式はブロック数の多い送信者の順番が後回しになる分，サブ転送で利用される可能性のあるブロックが増加するからである．

次に，トポロジ 12-3（図 8）でのシミュレーションの結果を表 4，表 5，表 6 に示す．

表 4　トポロジ 12-3 での各送信者の全転送完了時間

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S4	4.80	4.80	S0	1.60	1.60
S7	9.60	9.60	S4	6.40	6.40
S0	11.20	11.20	S7	11.20	11.20

表 5　トポロジ 12-3 での各送信者の平均転送完了時間

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S4	3.78	3.78	S0	1.60	1.60
S7	8.51	8.43	S4	5.40	5.40
S0	11.2	10.76	S7	10.18	9.89

表 6　トポロジ 12-3 での各送信者の転送ブロック数

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S4	8	8	S0	2	2
S7	8	13	S4	8	8
S0	2	5	S7	8	14

単純手法と提案手法を比較すると，表 4 より，各送信者の全転送完了時間は同じだが，表 5 より，A 方式では 2 番目の送信者から，B 方式では 3 番目 (最後) の送信者の平均転送完了時間が小さくなっていることが分かる．これはサブ転送を行った送信者から見た受信者の内いくつかには転送完了したが，全て（最後）の受信者へ転送完了した時間は単純手法と同じであったからだと考えられる．

また，表 6 より，トポロジ 5-5 と同様に，各送信者の転送ブロック数は単純手法よりも，提案手法の方が多くなっており，B 方式より A 方式の方が転送ブロック数が多くなっていることが分かる．この理由についてもトポロジ 5-5 と同様であると考えられる．

次に，トポロジ 12-4（図 9）でのシミュレーションの結果を表 7，表 8，表 9 に示す．

表 7　トポロジ 12-4 での各送信者の全転送完了時間

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S4	4.80	4.80	S0	1.60	1.60
S7	9.60	9.60	S11	3.20	3.20
S0	11.20	11.20	S4	8.00	8.00
S11	12.80	12.80	S7	12.80	12.80

表 8　トポロジ 12-4 での各送信者の平均転送完了時間

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S4	3.78	3.78	S0	1.60	1.60
S7	8.51	8.43	S11	3.20	2.91
S0	11.2	10.76	S4	6.98	6.80
S11	12.8	11.93	S7	11.78	11.49

表 9　トポロジ 12-4 での各送信者の転送ブロック数

A			B		
送信者	単純手法	提案手法	送信者	単純手法	提案手法
S4	8	8	S0	2	2
S7	8	13	S11	2	4
S0	2	5	S4	8	10
S11	2	6	S7	8	14

表 7, 表 8 より，単純手法と提案手法を比較すると，A 方式と B 方式ともに，各送信者の全転送完了時間は同じだが，2 番目の送信者から先は平均転送完了時間が小さくなっていることが分かる．この理由についてはトポロジ 12-3 と同様であると考えられる．

各送信者の転送ブロック数についてはトポロジ 5-5，トポロジ 12-3 と同様のことが言える．

また，以上の 3 つのトポロジを比較すると，送信者数が増えるほどサブ転送を行う機会が増え，サブ転送を行った送信者の転送完了時間が小さくなると分かる．加えて，送信者毎の全転送完了時間の平均，送信者毎の平均転送完了時間の平均という指標で見ると，3 つのトポロジのどれでも B 方式の方が小さい値になっている．これは，早く転送完了する送信者を優先した方が各送信者の平均は早くなるからである．

5. ま と め

本報告では，Coded-MPMC を基に，異なるファイルを持つ複数送信者からの一対多ファイル転送要求の発生時に効率的な同時転送を可能にする C-M2MT 手法の提案した．この手法は，複数の同時転送フェーズから構成され，全ファイルを適切な同一長のブロックに分割し，符号化が必要な送信者が符号化ブロックを生成後，各フェーズにおいて複数送信者から複数受信者へのブロック転送経路と各経路へのブロック割当を適切に決定する．

提案手法と単純手法のシミュレーションを 3 つのトポロジ上

で行い，比較した結果，送信者数が増えるほどサブ転送を行う機会が増え，転送完了時間を小さくすることができると示した．しかし，これは転送ブロック数とトレードオフの関係であることに注意する必要がある．また，今回用いたトポロジにおいて送信者優先度の決定方法は B 方式の方が良い結果が得られた．

今後の課題として，送信者順序の適応的な最適化，ファイルを分割するブロック長の最適化，大規模トポロジでのシミュレーション評価，OpenFlow 環境を用いた実機への実装などが挙げられる．さらに，より実用性を高めるためには，同時に発生する複数個の 1 対多ファイル転送のスケジュールだけでなく，異なる時間に順次発生する 1 対多ファイル転送要求列の適切なスケジュールに検討を進める必要がある．

文　　献

[1] L. Luo, H. Yu, et al., "Inter-Datacenter Bulk Transfers: Trends and Challenges," IEEE Network, vol. 34, no. 5, pp. 240–246, Sep. 2020.

[2] A. Nagata, Y. Tsukiji, et al., "Delivering a File by Multipath Multicast on OpenFlow Networks," Proc. the 5th International Conference on Intelligent Networking and Collaborative Systems, pp. 835–840, Sep.2013.

[3] M. Kurata, M. Shibata, et al., "Coded-MPMC: One-to-many Transfer using Multipath Multicast with Sender Coding," IEEE Access, vol.9, pp.49292-49307, Jan. 2021.

[4] 岡本洋平，他，複数経路マルチキャストを利用した多対多ファイル転送システムの試作，信学技法 CQ2016-11, 2016 年，4 月