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
CAS2021-75,CS2021-77(2022-03)

NDNによるキャッシュを用いたSFCにおけるファンクション選択手法

伊藤　　快†　　中里　秀則†

† 早稲田大学基幹理工学研究科情報理工・情報通信専攻　〒 169–8555 東京都新宿区大久保 3–4–1
E-mail: †konokono1117@fuji.waseda.jp, ††nakazato@waseda.jp

あらまし　IoT サービスを実現させるために，様々な機能 (ファンクション) を Service Function Chaining (SFC) という技術を使って Named Data Networking (NDN) 環境下で適用することを検討している．本稿では，負荷の分散や遅延を軽減した IoT サービスの実現を目的として，SFC のためのファンクション選択手法を提案している．提案手法では NDN の基本的なデータ構造である Forwarding Information Base (FIB) と Pending Interest Table (PIT) を拡張し，キャッシュの利用を考慮したうえでネットワーク全体で分散的なファンクション選択を可能とした．

キーワード　NDN，SFC，ファンクション選択問題，キャッシュ

Function selection method for SFC using cache in NDN

Konomu ITO† and Hidenori NAKAZATO†

† Department of Computer Science and Engineering，Waseda University　Okubo 3–4–1，　Shinjyuku-ku，
Tokyo，169–8555 Japan
E-mail: †konokono1117@fuji.waseda.jp, ††nakazato@waseda.jp

Abstract　We are investigating application of Service Function Chaining (SFC) to implement IoT services under the Named Data Networking(NDN) environment using．In this report，we propose a function selection method for SFC aiming to realize an IoT service with load distribution and reduced delay．In the proposed method，　The proposed function selection method enables distributed function selection while extending Forwarding Information Base (FIB) and Pending Interest Table (PIT)，which are the fundamental data structures for packet routing in NDN and taking the use of cache into consideration.

Key words　NDN，SFC，Function Selection problem，Cache

1. まえがき

現在，数多くの IoT サービスの普及に伴い IoT デバイスの数が急増している．IoT デバイスの増加に伴う通信トラフィックの増加により，クラウドやネットワークの負荷が増大し，ネットワーク内の遅延が大きくなっていくことが課題と考えられている．この課題を解決するために，本研究では Service Function Chaining (SFC) [2] という技術を用いることを提案している．SFC は IoT サービスの実現に必要な，様々な機能（ファンクション）を連携させることで多様なサービスを迅速かつ柔軟に構築したり，ネットワークの負荷の分散を図る技術である．本稿ではネットワーク内のルータにファンクションを配置することでファンクション処理による負荷を分散させることを提案している．なお，本稿においてファンクションを配置したルータを他のルータと区別するためにファンクションルータと呼ぶこととする．さらに本研究では SFC を Named Data Networking (NDN) [4] という名前ベースで通信を行うプロトコルで適用することを考えている．NDN における IoT サービスの SFC を実現させた関連研究はいくつか存在するが，本研究ではいずれの関連研究でも実施されていなかった，ファンクション選択を各ルータで段階的に行い，なおかつ NDN の標準的なデータ構造である Forwarding Information Base (FIB) と Content Store (CS) を用いたファンクション選択手法を提案する．

2. 関連研究

2.1 Named Data Networking

NDN はコンテンツ指向型ネットワークのプロトコルである．NDN における通信は，Interest パケットと Data パケットというコンテンツ名を情報として含む 2 種類のパケットをユーザとルータもしくはデータを保持するサーバ間を送受信させることで行っている．Interest パケットはコンテンツ要求のために用いられるパケットであり，ユーザが要求するコンテンツ名が記載されている．そしてこのコンテンツ名に基づいて，コンテンツ名を含むデータが存在するサーバに向かって送信される．

Data パケットはデータサーバやルータで取得したコンテンツをユーザへ向けて送るために用いられ，その中にもコンテンツ名が記載されている．

NDN の特徴として NDN ルータには，ルーティングやキャッシュデータの取得のために CS (Content Store)，PIT (Pending Interest Tables)，FIB (Forwarding Information Base) という 3 つのデータ構造が存在する．

CS: キャッシュ機能を実現させるデータ構造であり，Data パケットをキャッシュするためのデータストレージとなっている．NDN ルータに Data パケットが送られてくると，ルータはキャッシュポリシーに基づいて Data パケットをキャッシュするかどうかを判断した上で，パケットを CS にキャッシュする．キャッシュする際に CS にキャッシュできる空き容量がない場合，キャッシュリプレイスメントポリシーに基づいてキャッシュされているパケットのどれかを破棄し，新たなパケットのキャッシュを行う．

PIT: NDN では Data パケットの転送経路は，Interestp パケットが辿った経路を逆向きに辿っものとなる．Data パケットが Interest パケットが通過したルータをたどれるように，Interest パケットがどのユーザ，ルータから来たかを記憶するためのデータ構造が PIT である．

FIB: Interest パケットを，そこに記載されたコンテンツ名のコンテンツを提供するサーバまで送信するための，転送先を決めるデータ構造である．

NDN では上記の PIT，CS，FIB のデータ構造を用いることでユーザは要求するデータを受け取ることができるためこれら 3 つのデータ構造は NDN のアーキテクチャ上，通信を行うために重要な役割を担っている．

2.2 NDN-FC

NDN-FC は NDN における SFC を実現させた手法の 1 つである [3]．NDN-FC では Interest パケットに Function Name フィールドという，ファンクション名を記載できるフィールドを追加しており，その Function Name に記載のファンクション名に従ってパケット転送を行うことで SFC の実現を図っている．

2.3 DL-MD-C

NDN-FC では同じ種類のファンクションのインスタンスをネットワーク内に複数配備することを考えていなかったため，それを考慮したうえで SFC を実現した手法が Lord Distribution and Minimum Delay in Centralized (DL-MD-C) である [5]．DL-MD-C ではファンクション呼び出し回数と SFC 実行のために通過したホップ数を最小とするような評価関数に応じて各プレフィックスを識別子がついたものへ変更することで，同種のファンクションインスタンスのうち 1 つを選択することを可能としている．

2.4 Distributed Control

DL-MD-C では，評価関数を利用するためにすべてのファンクションの呼び出し回数とルータ間のホップ数といった情報を監視しておく集中制御の機能が必要となるが，実際のネットワークを想定するうえではではこの監視機能を実施することによって余計な遅延を生み出してしまうことが考えられる．そこで提案されたのが，ユーザと各ファンクションルータに情報を記載するテーブル構造を作成し，全体の監視を要せずにファンクションインスタンスの選択を行う手法である Distributed Control である [6]．Distributed Control では DL-MD-C でのファンクションインスタンス選択に必要な情報の取得をネットワーク全体で行い，ユーザだけでなくネットワーク内のファンクションルータで適切なファンクションインスタンスの選択を行うことを可能としている．

通常，NDN のアーキテクチャでは 2.1 でも述べた通り，経路制御に FIB を用いている．しかし，Distributed Control ではユーザとルータに追加したテーブルを使い経路制御を行っているため FIB を利用していない．そのため，本来の NDN の機能を有効に使えていない分，余計なデータ構造を作成してしまっているという問題点が挙げられる．加えてキャッシュ機能の利用を考慮していないためキャッシュ機能を利用できないという問題点も挙げられる．

3. 提 案 手 法

本稿では Distributed Control の経路選択に余計なデータ構造を用いている問題点とキャッシュ機能が考慮されていない問題点の解決のための，ファンクションの選択手法についての提案を行っている．

3.1 提案手法実現のための拡張

本提案手法実現のために，FIB の中に各ファンクションインスタンスの呼び出し回数と，各ファンクションノードまでのホップ数を記録するためのデータ構造を追加する．また，Interst パケット到着時に，この Interest パケットの次の送り先を示す FIB エントリへのポインタを記録しておくためフィールドを PIT エントリに追加する．さらに FIB に記録してある情報を更新するために Data パケットを拡張する．以下で順番に説明を行う．

3.1.1 FIB と PIT の拡張

まず NDN におけるルータが持つ元々のデータ構造である FIB に，呼び出し回数の記録のための Function Call Count (FCC)，ホップ数の記録のための Partial Hop Count (PHC) の 2 つのフィールドを追加する．また，PIT エントリに FIB のエントリのポインタを記録するための Selected Instance (SI) フィールドを追加する．SI は Data パケット転送時に FIB に追加した FCC と PHC の更新のために 1 つ手前のファンクションインスタンスの情報を確認するために用いる．

リクエストを満たすファンクションルータに Interest パケットが到着すると，まず、自身のファンクションインスタンスの FCC を 1 増加させる。尚、この自身の FCC は一定の時間によりリフレッシュされる。そしてファンクションを経由した後，FIB に追加した情報に応じてファンクション名の先頭プレフィックスのインスタンスを選択するが，選択した際に選ばれたインスタンスの呼び出し回数を 1 増加させる．その後，送り先の FIB のエントリが決定すると PIT に追加した SI にその FIB のエントリのポインタを記録する．図 1 に Interest パケッ

ト転送時の拡張した FIB と PIT の動きを示す．

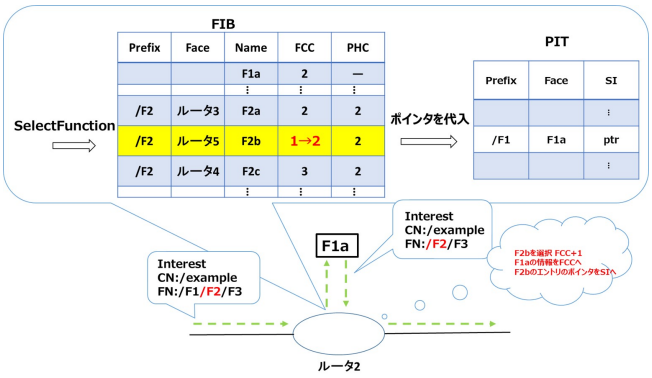

図 1　提案手法における FIB と PIT の役割 [7]

図のルータ 2 の FIB にはすでに他のファンクションインスタンスの呼び出し回数 FCC と呼び出し回数 PHC が記録されている．ここで最初にユーザが出したリクエストは/F1/F2/F3 という SFC を求めており，先頭プレフィックスに関してはすでに別のルータで F1a というファンクション F1 のインスタンスを選択済みのものとする．F1a のファンクションルータであるルータ 2 に Interst パケットが到着したため，F1a を経由し先頭プレフィックスの削除を行っている．そして再びルータ 2 に到達後 F1a の呼び出し回数 2 を FCC に記録している．次に先頭になった F2 のインスタンスの評価が行われ，F2 のインスタンス F2b が選択されて F2b の FCC を 1 追加し，送り先の FIB エントリを決定している．そして PIT の SI に FIB エントリのポインタを追加し，パケットを転送している．

3.1.2　Data パケットの拡張と FIB の更新

FIB に追加した情報を各ルータ間でやり取りをし，ファンクションの使用状況に応じて適切に更新するために Function Call Count フィールド (FCC フィールド)，Partial Hop Count フィールド (PHC フィールド) の 2 つのフィールドを Data パケットに追加する．FCC フィールドはファンクションインスタンスの呼び出し回数を記録するためのフィールドであり，PHC フィールドは Data パケットが転送されるファンクションルータ間のホップカウントを測定し記録するためのフィールドである．

2 つのフィールドの動きだが，FCC フィールドはサーバで Data パケットが生成されたとき，値を NULL に設定する．ファンクションルータに到達しファンクションインスタンスが実行され，Data パケットを転送するとき FCC フィールドにはそのファンクションインスタンスの呼び出し回数が代入される．PHC フィールドはサーバで Data パケットが生成されたとき，値を NULL に設定する．ファンクションルータに到達しファンクションインスタンスが実行され，Data パケットを転送するとき PHC フィールドには 0 が代入される．各ルータで Data パケットを転送するとき，PHC フィールドの値が NULL でない場合，このフィールドに 1 を加える．

FIB の更新の流れとしては，Data パケットが 1 つ目のファンクションルータに到達したとき，Data パケットの FCC フィールドと PHC フィールドは NULL であり更新するための情報を持たないため，FIB の更新は実行されない．2 番目以降のファンクションルータもしくはユーザと隣接するルータに Data パケットが到達したとき，Data パケットの FCC フィールドと PHC フィールドには FIB を更新するための情報が記載されているため PIT の SI に従い，ポインタ先のインスタンス名と一致するエントリを FIB の中から探索して，Data パケットに追加したフィールドに記載の情報をもとに FIB エントリを更新する．

Data パケット転送の流れを図 2 で示す．図 2 における Data

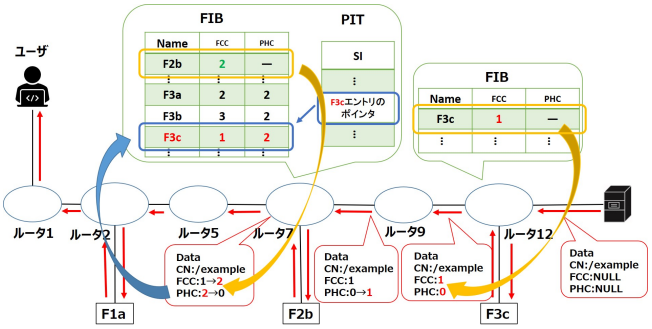

図 2　提案手法における Data パケットのフォワーディングプロセス [7]

パケットは F3c → F2b → F1a の順で転送される．Data パケットがはじめてファンクションルータに到達するルータ 12 では，まだ FIB のパラメータを更新するための情報を Data パケットが持っていないため，FIB の更新は行われない．そしてファンクションルータに到達したため，F3c 呼出し後 Data パケット転送の際に PHC フィールドは 0 に初期化される．その後 2 番目のファンクションルータであるルータ 7 に到着したときに PHC フィールドを 1 増やした後に FIB の更新が行われる．この時，PIT の SI に記録されているポインタを参照することで，1 つ前に実行したインスタンスが F3c だとわかる．そこで Data パケットの PHC フィールドが 2，FCC フィールドが 1 という情報を FIB の F3c のエントリに対して更新を行う．その後 F2b を経由後，F2b 自身の呼び出し回数である 2 を FCC フィールドに代入して，Data パケット転送時に PHC フィールドを 0 に初期化している．このような流れで FIB の情報の更新を実施している．

3.2　ファンクション選択アルゴリズム

本提案では [5] で用いられているファンクション選択のための評価関数を参考にし，FIB に記録されている情報からユーザや各ファンクションルータでファンクションインスタンスの選択を行う．つまり，ユーザが作成した Interest パケットが次の転送先であるルータに到達したとき，または Interest パケットがファンクションルータに到達したときに，そのルータ n_k は FIB を参照し，ファンクション名の先頭プレフィックスと一致するファンクションを検索する．そして FIB の各エントリからファンクションルータ間のホップ数 H_{kl} とファンクション呼び出し回数 C_{jl} を取得する．このとき，ルータ n_l はそのエントリが示すファンクションインスタンスが割り当てられている

ルータを表している．そして，式 (1) の S_l の値を最小化するルータ n_l が選択される．ここで，α と β はそれぞれ任意の定数である．

$$S_l = (\alpha H_{kl} + \beta C_{jk}) \tag{1}$$

3.3 キャッシュ機能の活用方法

本研究ではキャッシュ機能を有効にして，提案手法を実現させるためにキャッシュヒットした際の挙動を考えた．キャッシュヒットが起きた際にはそのキャッシュヒットが起きたルータから Data パケットが送信されるタイミングで FCC フィールドと PHC フィールドを NULL に設定する．その後は 3.1.2 と同様の流れで処理される．

これらの一連の流れを図 3 に示す．図 3 ではルータ 9 で

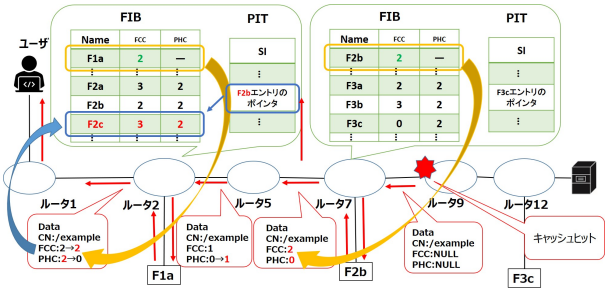

図 3　キャッシュヒット時の提案手法における Data パケットのフォワーディングプロセス [7]

キャッシュヒットが起きたこととしている．ルータ 9 から Data パケットが送信される際に Data パケットの FCC フィールドと PHC フィールドが NULL になっている．その後キャッシュヒットしてから 1 つ目のファンクションルータであるルータ 7 に到達するが，Data パケットには FIB を更新するための情報がないため更新は行わない．その後キャッシュヒットしてから 2 つ目のファンクションルータであるルータ 2 にパケットが到着する Data パケットは FIB を更新するための情報を持っているため PIT の SI を参照して FIB の F2b のエントリを更新している．そして F1a を経由後自身のファンクション呼び出し回数である 2 を FCC フィールドに代入し，ルータ 2 からルータ 1 へパケットを送信する際に PHC フィールドを 0 に初期化している．このような流れで，キャッシュヒットが起きた際も FIB の更新を行い，提案手法を維持できるようにした．

4. シミュレーション

提案手法を NDN のシミュレータである ndnSIM [1] を用いて評価した．比較するアルゴリズムとしては DL-MD-C，Distributed Control，Random Choice，Round Robin を用いた．Random Choice はファンクションインスタンスの選択をユーザがランダムで実施するアルゴリズムで Round Robin はファンクションインスタンスの選択をユーザがラウンドロビン方式で実施するアルゴリズムである．使用するネットワークトポロジーとしては図 4 に示す US24 トポロジー，図 5 に示す Sinet トポロジー，図 6 に示す Geant トポロジーを使用した．それぞれのネットワークにはユーザを 4 つ，サーバを 2 つを接続し，5 種類のファンクション (F1，F2，F3，F4，F5) のインスタンスをそれぞれ 3 つずつ，計 15 個のファンクションインスタンスを作成しネットワークのルータに接続する形で配置した．インスタンスを区別するための識別子には a，b，c を使用した．さらにキャッシュ機能の有効性を評価するためにこれらのアルゴリズムをネットワーク上の各ルータでキャッシュ機能を有効にした場合と無効にした場合で比較をし評価を行った．なお，キャッシュ機能利用時にはキャッシュポリシーには，ルータに到着した全ての Data パケットをキャッシュする NDN のデフォルトのキャッシュポリシーである CEE (Caching Everything-Everywhere) と，キャッシュリプレイスメントポリシーには最後に参照されてから最も長く時間が経過しているパケットを削除する LRU (Least Recently Used) を使用している．評価するための指標としては各ファンクションの呼び出し回数，ファンクション呼び出しの分散度，SFC 平均実行時間を用いた．

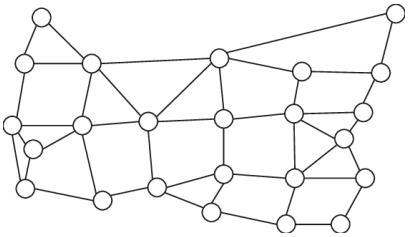

図 4　US24 トポロジー [7]

図 5　Sinet トポロジー [7]

5. 評　　価

まず，それぞれのトポロジーにおけるファンクション呼び出し回数のシミュレーション結果を図 7，8，9 に示す．末尾に C の記載があるものがキャッシュを利用したときの測定結果である．

どのネットワークにおいてもキャッシュを利用したときの方がファンクション呼び出し回数が少なくなっている．これは

図 6　Geant トポロジー [7]

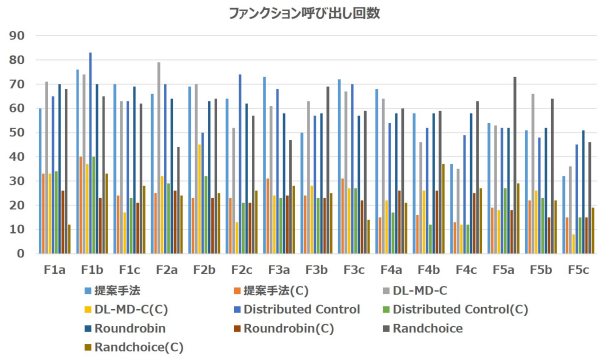

図 7　US24 トポロジーにおけるファンクション呼び出し回数 [7]

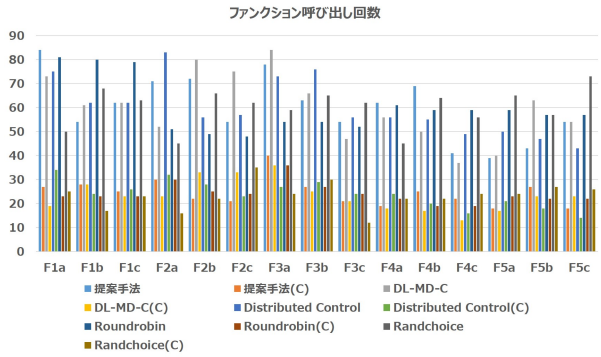

図 8　Sinet トポロジーにおけるファンクション呼び出し回数 [7]

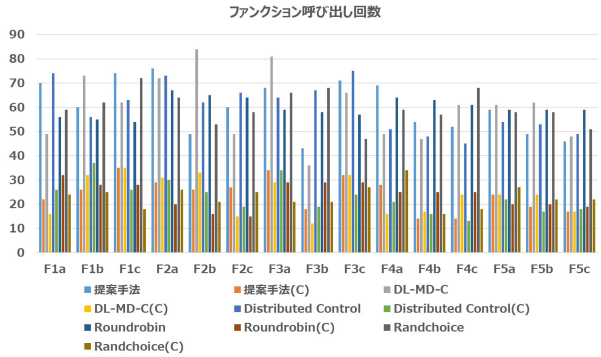

図 9　Geant トポロジーにおけるファンクション呼び出し回数 [7]

ファンクション呼び出し回数のカウントは Interst パケットがファンクションを通過する際にカウントされていくため，ネットワーク内のルータでキャッシュヒットが起きることにより，そのルータ以降に通るはずだった経路上にあるファンクションを通過する Interst パケットの数が少なくなったためだと考えられる．

次にファンクション分散度の結果を図 10，11，12 に示す．

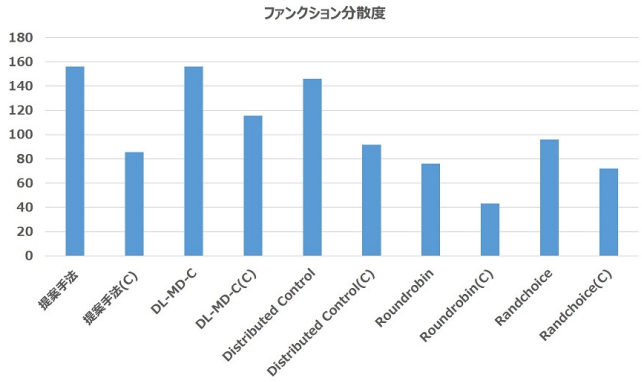

図 10　US24 トポロジーにおけるファンクション分散度 [7]

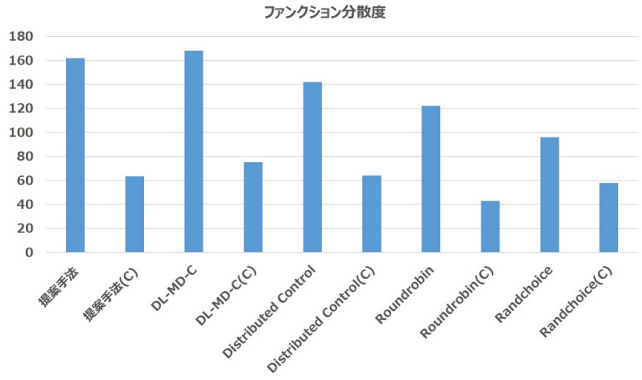

図 11　Sinet トポロジーにおけるファンクション分散度 [7]

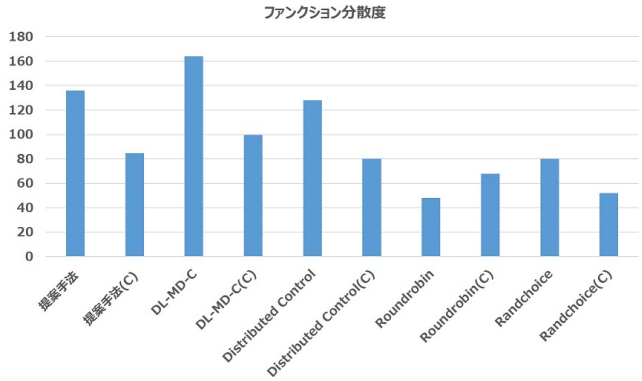

図 12　Geant トポロジーにおけるファンクション分散度 [7]

Random Choice と Round Robin ではキャッシュを利用した場合と利用しない場合にはあまり大きな違いは見られず，キャッシュによる影響は小さいものとみられる．一方，提案手法，DL-MD-C，Distributed Control においては分散度がどのネットワークにおいても大きく減少していることがわかる．

最後に SFC 平均実行時間の結果を図 13，14，15 に示す．今回のシミュレーションではリクエスト頻度を 10/s から 10/s 毎増やし，100/s まで測定した．

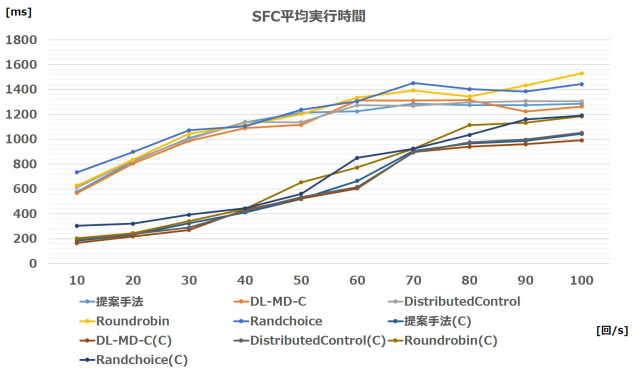

図 13　US24 トポロジーにおける SFC 平均実行時間 [7]

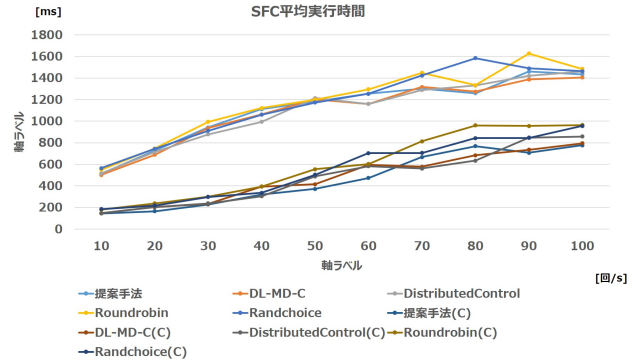

図 14　Sinet トポロジーにおける SFC 平均実行時間 [7]

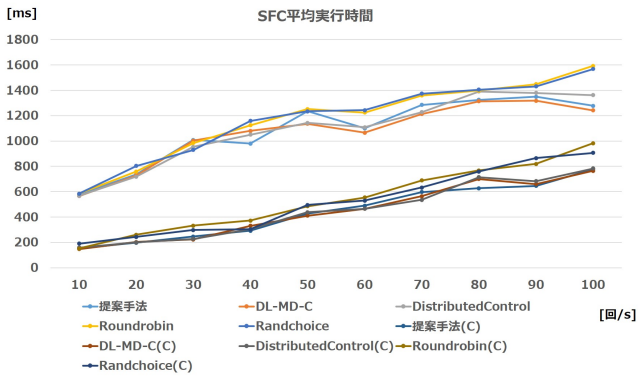

図 15　Geant トポロジーにおける SFC 平均実行時間 [7]

提案手法はキャッシュを利用する場合もしない場合も，DL-MD-C，Distributed Control と近しい結果となり，Random Choice，Round Robin と比べると SFC 実行平均時間が短いことがわかる．また，キャッシュを利用する場合と利用しない場合を比較するとキャッシュを利用した場合の方が大きく SFC 平均実行時間を短縮できていることがわかる．これはキャッシュヒットが発生したことにより，ファンクション処理の時間やキャッシュヒット後のルータへの転送時間が削減されたためと考えられる．

6. む　す　び

各ファンクションの呼び出し回数がキャッシュを利用した場合の方が少なかったこと，SFC 平均実行時間がキャッシュを利用した場合の方が低かったことから，NDN 環境下で SFC を実行しサービスを効率よく実現させるにはキャッシュ機能を用いることが有効であることがわかった．さらに提案手法に関しては，理想的なファンクション選択を行っていた DL-MD-C と大きな差異がない SFC 平均実行時間が得られた上に，Random Choice や Round Robin と比較すると SFC 平均実行時間の短縮が確認できたことから，IoT サービス実現のための効率の良い SFC 実行には提案手法が効果的であるといえる．そして Distributed Control とほぼ同程度の測定結果を得られたことから，本提案手法は NDN のアーキテクチャに即した分散制御型のファンクション選択手法であるといえる．

謝辞　本研究は総務省（平成 30 年度）戦略的情報通信研究開発推進事業（国際標準獲得型）【JPJ000595】の助成を受けたものです．

文　　献

[1] ndnSIM documentation. [Online]. Available: `http://ndnsim.net/current`.

[2] Joel M. Halpern and Carlos Pignataro. Service Function Chaining (SFC) Architecture. RFC 7665, October 2015.

[3] Yohei Kumamoto and Hidenori Nakazato. Implementation of NDN function chaining using caching for IoT environments. In *Proceedings of the Workshop on Cloud Continuum Services for Smart IoT Systems*, pp. 20–25, 2020.

[4] Named Data Networking Project. Named data networking project website. [Online]. Available: `https://named-data.net`.

[5] Yoshiaki Shiraiwa and Hidenori Nakazato. Function selection algorithm for service function chaining in NDN. In *2019 IEEE ComSoc International Communications Quality and Reliability Workshop (CQR)*, pp. 1–5, 2019.

[6] Naoki Yamaguchi and Hidenori Nakazato. Distributed control function selection method for service function chaining in NDN. In *Proceedings of the Workshop on Cloud Continuum Services for Smart IoT Systems*, CCIoT '20, p. 26–31, New York, NY, USA, 2020. Association for Computing Machinery.

[7] 伊藤快. NDN によるキャッシュを用いた SFC におけるファンクション選択手法. Master's thesis, 早稲田大学 基幹理工学研究科 情報理工・情報通信専攻, 2022.