

エンコードアドレスによるコンテンツ分散配置を実現する コンテンツセントリックネットワーク

北出 雄麻[†] 阿多 信吾^{††} 村田 正幸[†]

[†] 大阪大学 大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5

^{††} 大阪市立大学 大学院工学研究科 〒 558-8585 大阪府大阪市住吉区杉本 3-3-138

E-mail: [†]{y-kitade,murata}@ist.osaka-u.ac.jp, ^{††}ata@info.eng.osaka-cu.ac.jp

あらまし コンテンツセントリックネットワーク (Content Centric Network; CCN) は、コンテンツを主体として通信を行う新たなネットワークアーキテクチャである。CCN の利点の一つとして、コンテンツをルータでキャッシュすることによるネットワークの利用効率の向上がある。しかし、人気のあるコンテンツを多数持つ一部のノードにアクセスが集中し、その周辺経路のキャッシュが頻繁に更新される結果キャッシュの有効性が低下するという問題がある。本稿ではこの問題を解決するため、ランダムエンコードアドレスによりコンテンツを広域に分散配置する手法を提案する。提案手法についてシミュレーションによる性能評価を行い、特定ノードにコンテンツ要求が集中する場合に比べ、同一のキャッシュヒット率を達成するためのキャッシュ容量を最大 72% 削減できることを確認した。

キーワード 新世代ネットワーク, コンテンツセントリックネットワーク, エンコードアドレス, キャッシュ性能, 分散配置

On the Use of Encoded Addresses for Content Distribution in CCN

Yuma KITADE[†], Shingo ATA^{††}, and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology, Osaka University

1-5 Yamadaoka, Suita-shi, Osaka, 565-0871, Japan

^{††} Graduate School of Engineering, Osaka City University

3-3-138 Sugimoto, Sumiyoshi-ku, Osaka-shi, Osaka, 558-8585, Japan

E-mail: [†]{y-kitade,murata}@ist.osaka-u.ac.jp, ^{††}ata@info.eng.osaka-cu.ac.jp

Abstract CCN is considered as a future network architecture to exchange contents without specifying addresses of nodes. One of advantages to adopt CCN is to improve the availability of network resource by using in-network caching. However, since nodes having a lot of popular contents may suffer heavy traffic load which leads frequent updates of caches in neighbor nodes, the efficiency of cache usage may be degraded. To solve this problem, we propose a method to widely spread contents by using random encoded addresses. From our simulation results, we show that our method can reduce up to 72% of cache size to achieve the same cache hit ratio when the access frequencies are biased to a specific node.

Key words future network, CCN, encoded address, cache performance, distributed arrangement

1. はじめに

将来のネットワークアーキテクチャとしてコンテンツセントリックネットワーク (Content Centric Network; CCN) が近年検討されている。現在インターネットは、端末の識別子 (IP アドレス) を指定して通信する端末指向型の経路制御であるが、一方でユーザの利用形態はサービスやコンテンツを所望して通信をしているのが現状であり、経路制御と利用形態の間に乖離が存在する。CCN は、現状のコンテンツ指向型の通信形態と

親和性の高い、「コンテンツ」を主とした経路制御を実現することにより、通信形態の差異による問題の解消が期待されている。

CCN における利点の一つとして、コンテンツ情報自体をルーティングの対象とすることで、ノードの位置情報にとらわれない経路制御が実現できることが挙げられる。すなわち、コンテンツの複製 (コンテンツキャッシュ) を任意の中継ノードにおいて作成し、コンテンツを持つ最も近いノードへと経路制御することで、ネットワークの利用効率や応答性能の向上が期待される。コンテンツキャッシュは、単に性能の向上だけでなく、コンテンツを任意の場所に配置可能にすることにより、従来強い

結びつきのあったコンテンツとロケーションの結合を分離し、コンテンツの空間的・時間的制約を解消させる側面を持つ。すなわち、コンテンツキャッシュを複数配置することにより、オリジナルのコンテンツを持つノードがどこにあるか、さらにそのノードが稼働しているかどうかはコンテンツの取得において影響しない。

CCN の優位性を高めるためにはコンテンツキャッシュの効率的な生成および配置が重要である。特に CCN においてコンテンツは多様化・大容量化することが考えられるため、それらを効率的に発見し、またネットワーク資源の有効利用を実現するためには、コンテンツキャッシュの効率化が必須である。このため、これまでも CCN における効率的なキャッシュ手法に関する研究が進められているが、CCN に関する研究はまだ黎明期であることから、キャッシュ性能に関する評価についても基礎検討段階のものが多い。例えば、文献 [1,2] において、CCN におけるチャンクレベルのキャッシュ効果をシミュレーションにより明らかにしているが、使用しているネットワークトポロジは、シンプルなカスケードトポロジやツリートポロジに限定されている。

一方、近年ではより実モデルに近いキャッシュ性能の評価を行う研究がなされている。文献 [3] では、より一般的なネットワークトポロジにおけるキャッシュ性能について評価されている、さらに文献 [4] では、代表的な Tier-1 ネットワークトポロジに対して、コンテンツのアクセス頻度なども考慮した現実的なモデルによるキャッシュ性能の評価を行っている。

しかしながら、文献 [4] に代表される多くの評価では、コンテンツごとのアクセス頻度を Zipf 分布によりモデル化するなど、コンテンツに対するアクセス頻度の偏りについては検討されているが、ノードへのアクセス頻度の偏りについては考慮されていないものが多い。具体的には、ノードがどのコンテンツを保持しているかはランダムで与えられている。現実には、コンテンツごとのアクセス頻度と同様、コンテンツホルダーであるノードごとのアクセス頻度も不均一であることは想像に難くない。つまり人気のあるノード（サイト）は、人気のあるコンテンツを多く保持しているからこそ人気があるのであって、逆に言えばアクセス頻度の高いコンテンツほど特定のノードが保持している可能性が高いことは容易に想像できる。コンテンツの局所性が高い場合、アクセスの集中するノードおよびその周辺経路では頻繁にキャッシュの更新が発生し、キャッシュ利用効率が大幅に低下するものと考えられる。

そこで本稿では、CCN において、コンテンツをランダムに分散させることにより、コンテンツキャッシュの効率を高めるネットワークアーキテクチャの提案を行う。具体的には、コンテンツの検索およびルーティングについて、コンテンツ名自体を用いるのではなく、下位ルーティングプロトコルのアドレスングアーキテクチャに適応したランダムエンコードアドレスを用いる。またランダムエンコードアドレスが示すノードへのコンテンツの初期配置も行う。以上の内容についてシミュレーションによる性能評価を行い、キャッシュの有効性が向上することを明らかにする。

本稿の構成は以下の通りである。まず 2. において CCN の概要を説明し、3. で提案するコンテンツキャッシュ分散配置手法について述べる。さらに 4. でシミュレーションによる性能評価から提案手法の有効性について説明する。最後に 5. でまとめと今後の課題について述べる。

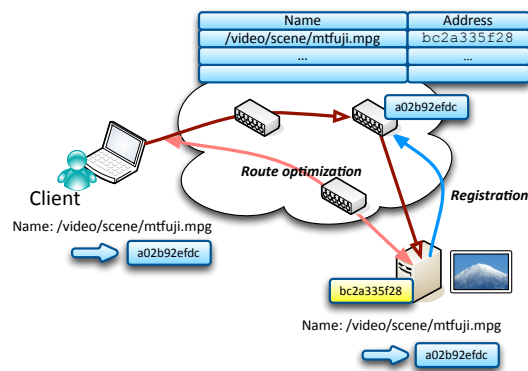


図 1 エンコードアドレスを用いた CCN の概要

2. CCN の概要

本章では、本稿で対象とする CCN の概要について説明する。CCN の研究は初期段階であり多くの実現アーキテクチャが提案されているが、本節では米国を主体に研究が進められている CCN/NDN [5] の概要について述べる。

CCN の通信は、コンテンツを要求する Interest パケットと、コンテンツを供給する Data パケットによって行われる。IP ネットワークにおいて、コンテンツは URL で参照されるが、CCN では階層的なコンテンツ名によって参照される。例えばりんごの画像ならば、"/picture/fruit/apple.png" のような名前を付けることが考えられる。

コンテンツによる経路制御を実現する CCN ルータは内部で Content Store (CS), Pending Interest Table (PIT), Forwarding Interest Base (FIB) の 3 種類の経路表を持つ。これらの経路表は Interest パケットが到着するごとに順に参照・更新される。まず、到着した Interest パケットに記述されたコンテンツ名にもとづき、CS が参照される。CS はすでにルータ内に存在するコンテンツキャッシュに関する情報を記載しており、Interest パケットで要求するコンテンツが CS に存在すれば、ルータの保持しているコンテンツキャッシュを返す。一方 FIB および PIT はそれぞれ Interest パケットの転送および Data パケットの返送に使用される。FIB はインターネットの経路表と同様に目的（コンテンツ名）に到達させるために転送すべき次ホップの情報が記載されている。CCN ルータは、Interest パケットに記載されたコンテンツ名をキーとして FIB を検索し、そこでマッチした次ホップの情報をもとに次のルータへと Interest パケットを転送する。加えて、Interest パケットに対する Data パケットをコンテンツ要求者に正しく届けるために、返送相手への転送先として PIT に情報を追加更新する。

3. コンテンツの分散配置を実現する CCN アーキテクチャ

本章では、コンテンツにランダム化されたエンコードアドレスを与え、エンコードアドレスによるコンテンツ配置とルーティングを行うことによってコンテンツの分散化を実現する CCN アーキテクチャを提案する。

3.1 エンコードアドレスを用いた CCN

図 1 に本稿で対象とするエンコードアドレスを用いた CCN の概要を示す。本稿では、コンテンツ名そのものをルーティングの対象アドレスとして考えるのではなく、下位レイヤのルー

ティング機構で使用されるアドレス空間にコンテンツ名をマッピング（本稿ではこれをエンコードと呼ぶ）することでルーティングを実現する。そして、下位レイヤのアドレス空間に適するようエンコードされたコンテンツ名のアドレスのことをエンコードアドレスと呼ぶ。

コンテンツ名ではなくエンコードアドレスを用いてルーティングを行うことのメリットとしては、(1) 下位のルーティング機構に大幅な変更を加えることなく CCN が実現できる、(2) エンコードアドレスのエンコード方法によって経路制御に付加的な機能を追加することが可能である、などがある。(1) は、すでにインターネットで運用されている、IPv4 あるいは IPv6 を下位ルーティングプロトコルとして用い、エンコードアドレスとして 128 ビットを使用した IPv6 アドレスを用いることで、実際のパケット転送は既存の IPv6 のルーティング機能を活用することが可能である。現有技術を最大限に活用することができれば、CCN への早期移行が大いに期待される。(2) は、例えばランダムにエンコードされたアドレスを使用することで、特別な処理なくコンテンツを広域に分散させることが可能であるなど、エンコード方法によってコンテンツの配置や経路制御などを管理することが可能であるということを表している。本稿におけるコンテンツ分散配置によるキャッシュの有効性向上は、このランダムエンコードを用いることによる分散を対象としている。詳細については 3.2 節で述べる。

図 1 では、例として下位ルーティングプロトコルのアドレス長が 40 ビットの場合を示している。ここでは、コンテンツ/video/scene/mtfuji.mpg という動画ファイルの取得を考える。まず、対象コンテンツを持つノード（コンテンツサーバ）は、自身がコンテンツを持つことを CCN に通知する。これを Registration と呼ぶ。

Registration はコンテンツのエンコードアドレスを用いて行われる。すなわち、コンテンツ名をエンコードしたアドレス a02b92efdc を宛先アドレスとして Registration メッセージを送信する。ここでアドレス a02b92efdc は、コンテンツ/video/scene/mtfuji.mpg のエンコードアドレスであると同時に、Registration の通知先であるノードの下位レイヤのアドレスという意味も持つ。したがって、同アドレスを宛先として送出された Registration メッセージは下位ルーティングプロトコルの経路制御に従って、ノード a02b92efdc に到達する。Registration メッセージには、コンテンツ名と共に、コンテンツサーバの下位レイヤのアドレス bc2a335f28 が含まれており、ノード a02b92efdc は、Registration メッセージを受信した段階で、コンテンツ名とサーバアドレスの対応表（コンテンツマッピングテーブル）にその情報を追加する。

一方、クライアントがコンテンツ/video/scene/mtfuji.mpg を取得したい場合、そのエンコードアドレス a02b92efdc を宛先とした Interest パケットを送信する。Interest パケットは下位ルーティングプロトコルの経路制御に従って、ノード a02b92efdc へと送られる。この時点で、Interest パケットを受け取ったノードは、メッセージに含まれるコンテンツ名をコンテンツマッピングテーブルから検索し、適合した場合に対応するコンテンツサーバのアドレスを取得し、Interest パケットをサーバに送信する。コンテンツサーバは Interest パケットを受け取った段階でクライアントに対してコンテンツを送信する。

以上の経路によりコンテンツが取得されるわけであるが、この場合コンテンツ/video/scene/mtfuji.mpg に関するすべてのパケットは、ノード a02b92efdc を経由することになるため、

通信経路上効率が悪い。このため、一度コンテンツサーバのアドレス bc2a335f28 が得られ、コンテンツの転送が開始された場合、クライアントとコンテンツサーバ間で直接通信することによって経路を短縮する方法が考えられる。これを経路最適化と呼ぶ。

3.2 ランダムエンコードアドレスによるコンテンツの初期分散配置

アクセス頻度の高いコンテンツが特定のノードに集中すると、その経路周辺でのキャッシュ更新が頻繁に発生することから、キャッシュの利用効率が低下することが考えられる。これを改善する方法として本稿では、エンコードアドレスにランダム性を持たせることで、コンテンツの初期配置ノードをランダムに分散させて決定する手法を提案する。

本稿では、上述のコンテンツの分散を簡便に実現可能な手法として、コンテンツ名に対してランダム性の高いエンコードアドレスを提供するランダムエンコードを用いる。ランダムエンコードとは、コンテンツ名に対して例えばハッシュ関数によってランダムなハッシュ値を計算し、そのハッシュ値をもとにエンコードアドレスを決定する方法である。SHA-1 などのランダム性の高いハッシュ関数ほど、ランダムエンコードとして適していると言える。

ランダムエンコードを行ったエンコードアドレスは、依然として下位レイヤのノードの物理アドレスとしての意味を持ち、さらにアドレスがランダム化されることによって、ネットワークのトポロジに依存せずランダムなノード位置を指し示すことが可能となる。従来、コンテンツの均一的な分散のためにはトポロジ構成を知る必要があったが、本稿ではランダムエンコードアドレスを用いることでランダムに選択されたノードのアドレスを設定し、コンテンツの分散配置を可能とする。

本稿では、3.1 節で示した CCN において、コンテンツを持つノード（コンテンツサーバ）は、コンテンツの存在をネットワークに通知（Registration）するのではなく、直接アドレス a02b92efdc を持つノードにコンテンツを初期配置する。これにより、クライアントは冗長な経路を経由することなく直接コンテンツを取得できるだけでなく、コンテンツ自体の分散配置が可能となる。

4. 提案手法の評価

本章ではランダムエンコードアドレスを使用してコンテンツをネットワーク上に分散配置した場合と、ランダムエンコードアドレスを使用せず人気のあるコンテンツを多数持つノードにアクセスが集中している場合を比較する。本稿ではキャッシュはコンテンツ単位ではなく、コンテンツを細分化したチャンク単位で行い、通信もチャンク単位で行われるとする。

4.1 評価環境

各手法の性能を計算機シミュレーションにより評価する。シミュレータとして ccnSim [4] を一部改変したものを使用する。具体的には、ccnSim ではコンテンツの人気度は考慮されているが、それらのコンテンツはネットワーク上のノードにランダムに分散されていることを仮定している。しかしながら本稿で対象とするような、人気コンテンツが特定ノードに集中する場合は想定されていないため、特定のノードに人気の高いコンテンツが集中するように ccnSim 内のコンテンツ配置アルゴリズムの修正を行い、ある特定のノードにコンテンツリクエストの 50%、70%、90% が集中している場合とランダムエンコードアドレスを使用してコンテンツが分散した場合の評価を行った。

コンテンツの人気分布は、Zipf の法則を用いて各コンテンツへの要求頻度を与えることで実現する。コンテンツリクエストの回数を f 、全コンテンツ中の人気の順位を r 、 c を定数、 α をパラメータとすると、Zipf の法則によるコンテンツの人気度は式 (1) で表される。

$$f_r = \frac{c}{r^\alpha} \quad (1)$$

コンテンツの人気分布はパラメータ α の値に大きく依存する。CCN のキャッシュ性能の評価に関する論文では、0.8 や 1.0 といった値が使われている [6–8]。また、コンテンツ投稿型サービスである DailyMotion は $\alpha \approx 0.88$ であることが文献 [9] で示されている。本稿では α の値として、0.9 を使用する。

また、Interest パケットの発生は 1 秒あたり平均 100 個のポワソン分布に従うとし、個別コンテンツに対するリクエストの頻度は上述の Zipf の法則による分布に従うものとする。ただし、自身が持つコンテンツへのリクエストは発生しないものとする。また、直近のリクエストに対するコンテンツ取得が完了していない場合、同一のリクエストは発生しないものとする。

ランダムエンコードアドレスを使用する場合、使用しない場合ともに、各ノードにキャッシュしていない Data チャンクが転送されてきた場合、常にキャッシュすることとし、キャッシュサイズ以上のコンテンツキャッシュが発生した場合のキャッシュ置き換えアルゴリズムは LRU (Least Recently Used) を使用する。

シミュレーションの評価は、十分にシミュレーションの結果が安定した段階で行う。すなわち、全てのノードのキャッシュが一杯になり、全てのノードのキャッシュヒット率が収束してから評価指標を計算する。キャッシュヒット率が収束してからさらに一定時間経過した段階でシミュレーションを終了し、これを乱数のシードを変化させて 5 回行い、その平均値を評価結果とする。

従来手法であるエンコードアドレスを使用しない場合と、提案手法であるエンコードアドレスを使用する場合について、それぞれルータノードのキャッシュサイズ (各 CCN ノードが格納できるデータチャンク数) を変化させて比較する。またコンテンツ数は 10^7 、平均チャンク数は 100 とする。

また各ノードへの経路は、ホップ数による最短経路があらかじめ求められているものとし、シミュレーション中経路変化はないものとする。

評価トポロジとしてノード数 46 の Level3 トポロジを使用する [10]。ただし、データとして使用したネットワークトポロジは図 2 に示すとおりコアネットワークのみであるため、エンドノードが接続された状態を考慮し、基幹ノードごとに 3 個の端末ノードが接続されたものを使用する。その結果、ノード数 184、直径が 6 となる。またサーバ、クライアントは端末ノードに存在するものとする。

また、本稿ではキャッシュの利用効率を主対象としていることから、全てのノードがキャッシュを持つと仮定し、回線容量は十分大きく、輻輳は発生しない条件下で評価を行う。

以下、シミュレーションにおいては、初期配置された (キャッシュではない) コンテンツを持つノードをリポジトリと呼ぶ。

4.2 評価指標

評価指標として、キャッシュヒット率とホップ数短縮率を使用する。キャッシュヒット率とは、リポジトリへの経路上でキャッシュにヒットする確率である。つまりノード $1, 2, \dots, n$ が存在し、ノード n をリポジトリとした場合、ノード i でキャッシュ



図 2 Level3 トポロジ

にヒットした回数を H_i とすると、キャッシュヒット率 Q は式 (2) で表される。なお、本稿では Interest パケットを生成したノードにおいてもまずキャッシュを確認し、ヒットしたかどうかを判断する。Interest パケットがリポジトリまで到達した場合はノード n でヒットしたとする。

$$Q = \frac{\sum_{i=1}^{n-1} H_i}{\sum_{i=1}^n H_i} \quad (2)$$

ホップ数短縮率とは、Interest パケットを生成したノードに要求した Data パケットが転送されてきたときの、Data パケットが実際に経由したホップ数 d を、Interest パケットを生成したノードと求める Data パケットを持つリポジトリとの最短ホップ数 P で割った値の平均値である。例えば、ノード $1, 2, \dots, n$ が存在する場合、ノード i および j 間の最短ホップ数を $H_{i,j}$ 、ノード k が生成したコンテンツ c への Interest パケットに対してノード l のキャッシュ (リポジトリ) が使われたときに、その回数を $D_{c,k,l}$ とする。また、コンテンツ c のリポジトリを r_c とする。ノード i が生成したコンテンツ c への Interest パケットの総数を $I_{c,i}$ 、コンテンツ c への Interest パケットの総数を I_c 、Interest パケットの総数を I とすると、

$$I_{c,i} = \sum_l D_{c,i,l} \quad (3)$$

$$I_c = \sum_i I_{c,i} \quad (4)$$

$$I = \sum_c I_c \quad (5)$$

となり、ノード i が生成した、あるコンテンツ c に対する Interest パケットについて、ノード j のキャッシュ (リポジトリ) が使われた場合、一つの Interest パケットに対するホップ数短縮率を

$$p_{c,i,j} = \frac{H_{i,j}}{H_{i,r_c}} \quad (6)$$

とすると、ノード i があるコンテンツ c に対して Interest パケットを出したときのホップ数短縮率の平均は式 (7) で表される。

$$p_{c,i} = \frac{\sum_j p_{c,i,j} D_{c,i,j}}{I_{c,i}} \quad (7)$$

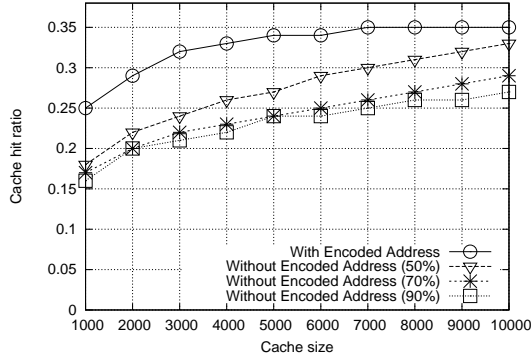


図3 エンコードアドレスを用いた場合と用いない場合のキャッシュヒット率の比較

また、あるコンテンツ c に対するホップ数短縮率の平均は式 (8) で表される。

$$p_c = \frac{\sum_i p_{c,i}}{N} \quad (8)$$

求めるホップ数短縮率の平均値は式 (9) で表される。なお C はコンテンツ数とする。

$$P = \frac{\sum_c p_c}{C} \quad (9)$$

式 (9) を計算し、整理した式を式 (10) に示す。

$$P = \frac{I}{NC} \sum_c \sum_i \sum_j \frac{H_{i,j}}{H_{i,r_c} I_{c,i}} \quad (10)$$

この値が 1 以下ということは、リポジトリへの経路上のノードで求める Data パケットを見つけたことになり、この値が小さいほど応答時間の短縮が実現できていることになる。

4.3 評価結果

図3から図6において、“without Encoded Address”に付加された割合は、アクセスの集中するノードにコンテンツリクエストの何パーセントが集中しているのかを示す。すなわち 90% であれば、全リクエストの 90% が単一ノードに集中していることを示す。なおキャッシュサイズはコンテンツ単位とする。すなわちキャッシュサイズ 2000 は、各ノードがコンテンツ 2000 個分のキャッシュサイズを持つという意味である。以下、コンテンツ単位のキャッシュサイズを S とする。

図3, 4より、エンコードアドレスを用いた手法によってキャッシュヒット率、ホップ数短縮率ともに改善していることがわかる。これはエンコードアドレスを用いてコンテンツをランダムなノードに初期配置したことにより、アクセスの集中するノード付近のキャッシュの置き換わりの多発がなくなったためであると考えられる。具体的にはエンコードアドレスを用いた手法により、ある特定のノードにコンテンツリクエストの 50% が集中している場合に比べ、キャッシュヒット率が最大 8%、コンテンツリクエストの 70% が集中している場合に比べ最大 10%、コンテンツリクエストの 90% が集中している場合に比べ最大 11% 改善している。ホップ数短縮率はいずれの場合も約 3% 改善している。文献 [6] は、本稿の評価トポロジと同規模のスケールフリートポロジを用いて評価を行っているが、キャッシュヒット率、ホップ数短縮率ともに約 3% の改善であり、本稿の

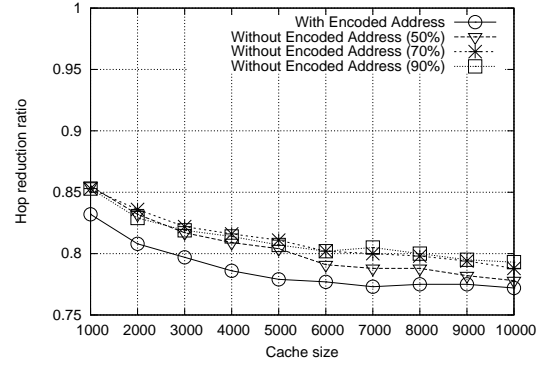


図4 エンコードアドレスを用いた場合と用いない場合のホップ数短縮率の比較

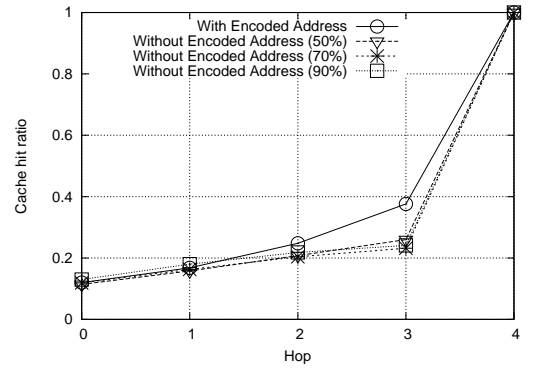


図5 エンコードアドレスを用いた場合と用いない場合のクライアントからのホップ数ごとのキャッシュヒット率の比較 ($S = 5000$, $h = 4$)

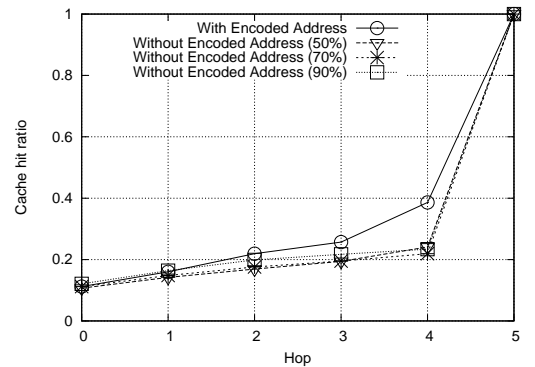


図6 エンコードアドレスを用いた場合と用いない場合のクライアントからのホップ数ごとのキャッシュヒット率の比較 ($S = 5000$, $h = 5$)

提案手法においても十分な性能向上があるとみなすことができると考える。

ホップ数短縮率の向上がどのような理由により達成されているかをより詳細に分析するため、要求したコンテンツがリポジトリへの経路上のどのノードのキャッシュによりアクセスされているかを考える。このため、クライアントとリポジトリの最短ホップ数が h としたとき、クライアントからのホップ数ごとのキャッシュヒット率をそれぞれ求める。まず、最短ホップ数 h のリポジトリへの Interest パケット数を R_h 、最短ホップ数 h のリポジトリへの Interest パケットに対応するコンテンツが、クライアントからホップ数 i のノードのキャッシュで見

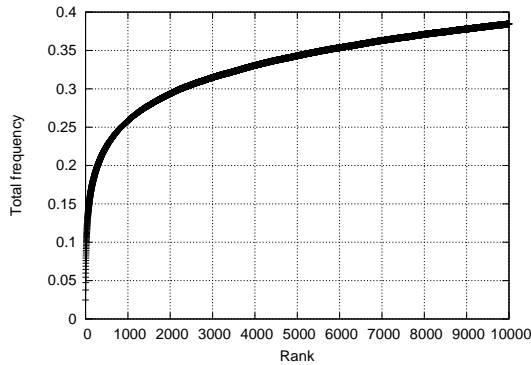


図7 人気上位コンテンツの個数ごとのコンテンツリクエストを占める割合 (コンテンツ数 10^7 , $\alpha = 0.9$)

つかった回数を $E_{h,i}$ とする. なおリポジトリからコンテンツを取得した場合はホップ数 h のノードのキャッシュで見つかった (すなわち回数は $E_{h,h}$) とする. このとき, クライアントからホップ数 i のノードまでのキャッシュヒット率 $C_{h,i}$ は式 (11) で与えられる.

$$C_{h,i} = \frac{\sum_{j=0}^i E_{h,j}}{R_h} \quad (11)$$

キャッシュサイズ 5000 (コンテンツ単位) のとき, エンコードアドレスを用いた場合と用いない場合のクライアントからのホップ数ごとのキャッシュヒット率の比較が図 5, 6 である. これらの図において, グラフの横軸は i , 縦軸は $C_{h,i}$ を表している. また, 図 5, 6 の h はそれぞれ $h = 4, 5$ である.

図 5, 6 から, エンコードアドレスを用いた手法により, 特に $h - 1$ ホップ目のキャッシュヒット率が $h = 5$ の場合においておよそ 2 倍に向上していることが分かる. これは, 特定ノードへの人気コンテンツの集中によって, 特定ノード周辺のキャッシュ利用率が大幅に低下していたものが, エンコードアドレスによるコンテンツの分散の結果, キャッシュの置き換え頻度が低下したためであると考えられ, 提案手法によって問題が解決されていることを表しているといえる. ただし全体で見た場合ホップ数短縮率が 3% に留まっているのは, ホップ数の異なる他のノードにおけるキャッシュヒット率に変化がないためであり, これはもともとクライアントに近いノードではキャッシュの置き換えがあまり発生しないことによるものである.

図 3, 4 より, キャッシュサイズ 5000 以上において, エンコードアドレスを用いた手法におけるキャッシュサイズを大きくすることの影響が小さくなっていることがわかる. これは zipf 則による人気の偏りが原因である. 図 7 はコンテンツ数 10^7 , zipf 則のパラメータ $\alpha = 0.9$ のときの人気上位コンテンツの個数ごとのコンテンツリクエストを占める割合である. この図より人気上位 5000 個のコンテンツはコンテンツリクエストの約 34%, 人気上位 6000 個のコンテンツはコンテンツリクエストの約 35% を占めることがわかる. つまりキャッシュサイズを 5000 から 6000 まで大きくすることはキャッシュヒット率の改善にほとんど繋がらず, キャッシュサイズ 5000 以上においてキャッシュサイズ 1000 の差は性能にほとんど影響を与えない.

また図 3, 4 より, キャッシュサイズが 5000 以上の領域においては, 特定ノードにより処理されるリクエスト数の割合が高い (70, 90%) ほど, キャッシュサイズ増加によるヒット率の向

上が限定的になっていることが分かる. 一方 50% のリクエストが特定ノードに集中する場合は, 依然としてキャッシュサイズの増加によってヒット率が向上している. これは, リクエストの集中により特定ノード周辺のキャッシュの置き換え頻度が増加してキャッシュ利用率が低下するだけでなく, 他のノードではキャッシュ自体がすべて活用されていないという利用率の低さという 2 つの要因が起こることにより, キャッシュの利用率改善が非常に限定的となっていると考えられる.

5. 結論と今後の課題

本稿では, CCN におけるランダムエンコードアドレスを使用したコンテンツ分散配置手法の提案とその評価を行った. この手法を用いることにより, コンテンツをネットワーク上に簡単に分散配置でき, キャッシュ性能が向上することを確認した. さらに, 既存の下位ルーティングプロトコルを用いてルーティングを行うので, CCN への早期移行も期待できる.

本稿ではキャッシュしていないコンテンツを持つ Data パケットが転送されてきたときに常にキャッシュするとしたが, さらなるキャッシュ性能向上のために新たなキャッシュアルゴリズムを考える余地があると考えられる.

謝 辞

本研究成果の一部は総務省・戦略的情報通信研究開発推進制度 (SCOPE) の支援による.

文 献

- [1] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," Proceedings of 23rd International Teletraffic Congress (ITC 23), pp.111–118, Sept. 2011.
- [2] G. Carofiglio, M. Gallo, and L. Muscariello, "Bandwidth and storage sharing performance in information centric networking," Proceedings of ACM SIGCOMM Workshop on Information-Centric Networking (ICN-2011), pp.26–31, Aug. 2011.
- [3] E.J. Rosensweig, J. Kurose, and D. Towsley, "Approximate models for general cache networks," Proceedings of IEEE INFOCOM 2010, pp.1–9, March 2010.
- [4] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," Technical report, Telecom ParisTech, July 2011.
- [5] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard, "Networking named content," Proceedings of ACM CoNEXT 2009, pp.1–12, Dec. 2009.
- [6] W.K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "Less for More" in Information-centric Networks," Proceedings of Networking 2012, pp.27–40, May 2012.
- [7] I. Psaras, W.K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," Proceedings of ICN'12, pp.55–60, Aug. 2012.
- [8] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," Proceedings of IEEE NOMEN2012, pp.310–315, March 2012.
- [9] Y. Carlinet, B. Kauffmann, P. Olivier, and A. Simonian, "Trace-based analysis for caching multimedia services," Technical report, Orange labs, 2011.
- [10] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," Proceedings of ACM SIGCOMM 2002, pp.133–145, Aug. 2002.