

# 近接キャッシュコンテンツを探索する Local Tree Hunting 方式

清水 洋 地引昌弘

CCN/ICN はコンテンツを網内キャッシュという特徴を有し注目を浴びている。本論文では、部分的探索でも全探索に匹敵する発見能力を有する近接キャッシングノードの探索方式を提案する。これは、キャッシング履歴を有する近接ノードを Root とする Local Tree を探索範囲とし、さらに、キャッシング情報を隣接ノードに通知することで、要求信号をショートカットし、探索オーバーヘッドを削減することにより実現される。

## 1 はじめに

従来、ネットワークは情報の伝達手段として設計されていたのに対し、最近は色々な情報処理機能、例えば計算機能や蓄積機能を有するネットワークが、Future Internet として期待を集めている。中でも、コンテンツを中継するノードがそれをキャッシュする機能を有した Information Centric Network (ICN) が、脚光を浴びている<sup>[1]-[6]</sup>。従来の方式ではコンテンツが Web サーバからダウンロードされていたのに対し、ICN ではユーザに近いキャッシングノードからダウンロードされるので、レスポンス時間の短縮や中継伝送リンク (ホップ数) の削減を図ることができる。したがって、ICN ではコンテンツをどのように転送するかよりも、キャッシュされている所望のコンテンツをどのように探すかが重要な課題となる。現状では、Domain Name Server (DNS) において、コンテンツを格納する Web サーバの IP アドレスを、コンテンツ名 (例えば、URL アドレス) から解決している。しかしながら、ICN では、コンテンツは、ネットワーク内のノードに対し広く頒布されると共にダイナミッ

クにキャッシュ及び廃棄 (Eviction) されるので、DNS が集中的にコンテンツの在り処を管理することは、困難である。

本稿は、このような状況に鑑み、集中管理ではなく、分散的に所望コンテンツを探索・発見する手法を提案する。コンテンツの要求者は、コンテンツの所在が不明なので、場所を示す IP アドレスではなくコンテンツ名で要求を行い、ネットワークは、コンテンツをキャッシュしているノードまでこの要求を導く。ここで重要なのは、レスポンスタイムの短縮・中継伝送リンク数の低減であり、要求ノードに近接しているキャッシングノードを発見することが望まれる。つまり、Internet では最短経路を与えるのに対し、ICN では、コンテンツの近接キャッシングノードを与えることが重要となる。

ICN の実現形態の 1 つとして、Content Centric Network (CCN)<sup>[2][6]</sup> があり、その特徴を、図 1 を用いて説明する。最も大きな特徴は、ネットワークノードはコンテンツをキャッシュする機能を有し、コンテンツ名を用いた要求信号により最寄りのキャッシングノードを探索するという点である。ユーザ R1 は、あるコンテンツの要求信号を送出する。この Interest と称する要求信号は、コンテンツ名が特定する経路、Name base path (黒実線) に沿って転送され、Web Server に至る。Web Server からダウンロードされるコンテンツは、Interest がたどった経路とは逆方向 (黒破線) に転送され、中継ノード A、B、C はそれをキャッシュする。コンテンツ名は IP アドレスと同様、Prefix 構造を有し、コンテンツ名による経路制御を可能としている。次に、ユーザ R2 が同じコンテンツを要求すると、この Interest は、ユーザ R2 を端点とする Name base path (赤実線) に沿って転送され、最寄りのキャッシングノード B に至る。ノード B は、コンテンツのダウンロード元 (赤破線) となり、Web

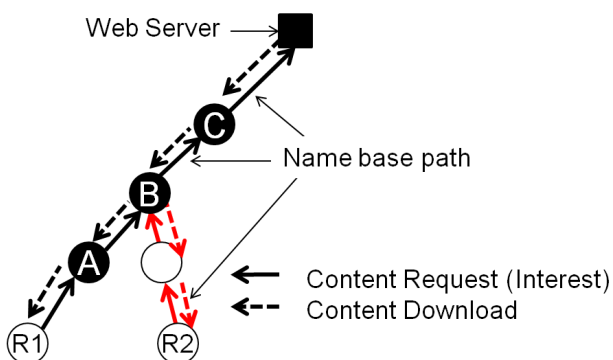


図 1 CCN 方式

Server からダウンロードする従来方式に比べ、転送経路を短縮することができる。しかしながら、この方法では、必ずしも近接のキャッシングノードが探索されない、という課題を有している。一方、近接のキャッシングノードを探索するため要求信号を Flooding する方法<sup>[6]</sup>も検討されているが、網羅的な探索のためオーバーヘッドが大きくなるという課題がある。本論文では、Local Tree Hunting (LTH) と称する、部分的探索でも全探索に匹敵する探索、すなわち、最近接キャッシングノードを発見する分散制御型の探索方法を提案し、その有効性を計算機シミュレーションにより検証する。**2**では、従来のコンテンツ探索方式を紹介し、**3**では、提案する LTH 方式の詳細な説明を行い、**4**では、探索の為のアクセスプロトコル、**5**では、比較対象となる方式の紹介及び計算機シミュレーションにより性能比較・評価を行い、本提案方式の有効性を示す。

## 2 従来の CCN/ICN 方式

従来の代表的な CCN/ICN 方式について説明する。

1) CCN 方式 (図 2(a))<sup>[7]</sup>: ノード R からのコンテンツの要求は、Name base path 上での直近のキャッシングノード C (Default caching node) において終端され、ノード C から逆の経路をたどって逆の方向にダウンロードされる。要求信号によるキャッシングノードの探索範囲は、Name base path 上に限定されるので、探索のためのオーバーヘッドを小さく抑えることができるが、図に示す様に、Name base path 上にはないキャッシング

ノード、例えば、ノード C より近接にあるキャッシングノード 1、2 は発見できない。

- 2) Breadcrumb 方式 (図 2(b))<sup>[7]</sup>: 過去のコンテンツの転送した経路に履歴を示す Breadcrumb<sup>[8]</sup>を撒き、それに従って要求信号を転送する方式である。ノード A は過去にコンテンツを格納し、転送した履歴を有している場合、そのコンテンツが Eviction された後も、その履歴を Breadcrumb として保持する。図の場合、ノード R からの要求信号は、ノード A の Breadcrumb に従い、ノード 1 の方向に転送される。Breadcrumb は上書きされ、最新の痕跡 (Most Recent Trail) が記録される。この方式は、Name base path 上での直近のキャッシングノード C より、過去にキャッシュした履歴を有するノードの周辺にあるキャッシングノードの方が要求ノードに近いという見込みを前提としており<sup>[7]</sup>、発見されたキャッシングノードが、ノード C より近いという保証はない。
- 3) Flooding 方式 (図 2(c))<sup>[6]</sup>: コンテンツ要求は全ての隣接ノードに Flooding され、この処理がキャッシングノードに到達するまで繰り返される。キャッシングノードは、要求ノードに対し応答信号を送出し、要求ノードは、応答信号をチェックして、最近接にあるノードを特定し、ダウンロードを指示する。Name base path とは関係なく、網羅的に探索するので、最短にある近接ノードが発見されるが、探索のためのオーバーヘッドが大きい。

## 3 Local Tree Hunting (LTH) 方式

本方式は、時間経過と共にコンテンツのキャッシュ範囲は広がるという特性を踏まえ、網羅的に探索しなくても、十分最近接のノードを発見する手法である。図 3、4 を参照しながら基本方式及び付加方式を説明する<sup>[8][9]</sup>。

### 3.1 基本方式

LTH 方式として LTH (S) と LTH (M) の 2 通りがあり、初めに、図 3 (a) に示す LTH (S) 方式を説明する。要求ノード R は、図 2 (a) に示す CCN 方式と同様、コンテンツ名を含む要求信号を、Web Server に至る Name base path に送出する。このコンテンツ要求は、コンテンツをキャッシュしていないノードでは中継され、コンテンツをキャッシュしている最初のノード (Default Response Node と呼ぶ) がこれに応答し、応答信号を要求ノード R に対して返送する。同時に、この要求信号を、このノードを Root とする

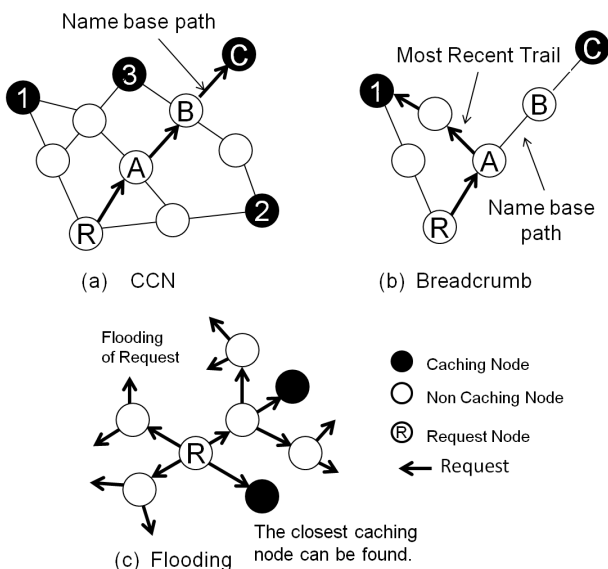


図 2 従来方式

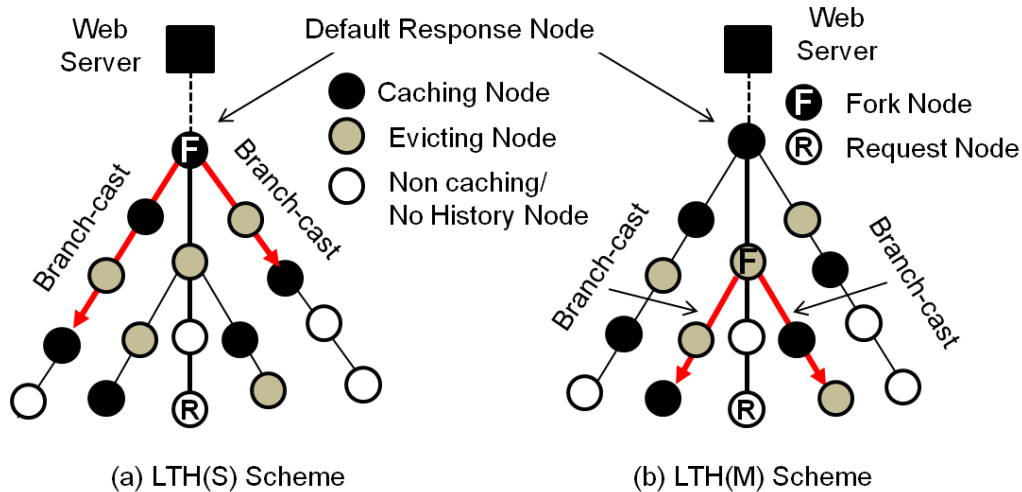


図3 Local Tree Hunting 基本方式

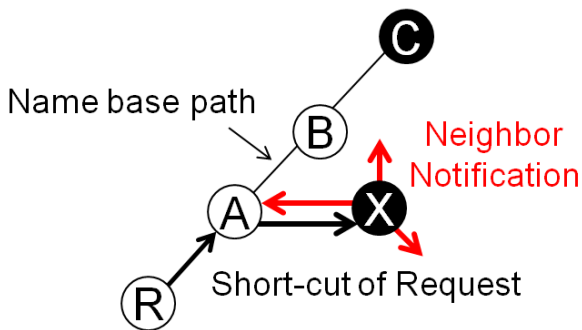


図4 Neighbor Notification 制御

Local Tree 内の転送履歴のある全ての経路に Branch-cast する (Root となり Branch-cast するノードを Fork Node と呼ぶ)。要求信号を受信したキャッシングノードは、要求ノード R に対し、応答信号を送出する。キャッシュした履歴はあるもののコンテンツが Eviction されたノードでは、要求信号を履歴に従って転送する。要求ノード R は、受信した応答信号の中から、最もホップ数の小さいものを選択し、送信元のキャッシングノードに対して、Acknowledgment を送出する。以上の手順により、応答ノードのうち、要求ノードに最も近いキャッシングノードが発見される。

CCN 方式との違いは、1) Fork Node が、要求信号を Local Tree 上の転送履歴のある経路に対して Branch-cast すること 2) 要求信号を受信したキャッシングノードからの応答信号に基づき、要求ノードにおいて、近接ノードを選択すること 3) 応答信号の転送及びコンテンツのダウンロードは、要求ノードの Location Address を用い、最短経路をたどることの3点である。

次に、LTH (M) 方式を、図 3 (b) を用い説明する。Name base path 上で、要求ノードに最も近いキャッシングノード (Default Response Node) が応答するという点は LTH (S) と同じだが、キャッシング履歴

を持つ近接ノードが Fork Node となり、Branch-cast する点異なる。但し、Fork Node がコンテンツをキャッシュしている場合は、Default Response Node が Fork Node となり、LTH (S) と同じ振る舞いをする。

LTH (M) は、LTH (S) に比べ要求ノードに近いノードが Branch-cast するので、Local Tree のサイズが小さく探索範囲は狭い。一般に探索範囲が広いほど、より近接するキャッシングノードが発見できるので、探索範囲の広い順、すなわち、各方式は、Flooding/ LTH (S) / LTH (M) / CCN の順で発見能力が高いと言える。

### 3.2 付加的制御

探索能力を向上すべく、図 4 に示す Neighbor Notification 制御<sup>[10]</sup>を付加する。新たにコンテンツをキャッシュしたノード X は、その情報を隣接ノードに通知する。一方、通知を受けたノード A は、そのコンテンツの要求信号を受信すると、Name base path に沿ってノード B には転送せず、コンテンツの存在が確実な隣接ノード X にショートカットする。ノード X は、Fork Node として機能し、Branch-cast を実施する。Neighbor Notification 制御自体、新たな探索オーバーヘッドとなるので、1) 要求ノード及び中継ノードが実施する場合 NN (AL) と 2) 探索オーバーヘッドを小さくすべく要求ノードのみが実施する場合 NN (RQ) とを評価する。

## 4 アクセスポトコル

LTH プロトコルは、図 5 (a) に示すように Content Request, Response, Acknowledgment, Content Download の4つのフェーズから構成され、各フェーズは、図 5 (b) に示すフレームのヘッダにある Type

フィールドで識別される。各ノードには、Location を示す Node ID が割り当てられている。要求ノード R は、Content Name (例えば、URL アドレス) に、一時的に対応付けられた Content Label を生成する。要求コンテンツは、Content Label と Request Node ID のセットによりユニークに定義される<sup>[11]</sup>。同種の試みが ICN/SDN の検討の中で行われているが、これらの試みでは Content Label が集中的に管理される<sup>[12]</sup>のに対し、本方式では各ノードが分散的に管理している。

図5のアクセスプロトコルを、4つのフェーズに分けて以下に説明する。

- 1) Content Request : コンテンツ要求は、Fork Node (F) を Root とする Local Tree 内で Branch-cast される。各ノードは、Content Name に基づき履歴のチェック・転送制御を行う。
- 2) Response : Local Tree 内のキャッシングノード (Fork Node 及び Response Node 1、2、3) は、宛先を Request Node ID とする応答信号を送出する。応答信号は、Request Node ID が与える最短経路を用いて転送される。
- 3) Acknowledgment : Request Node は、キャッシングノードからの応答信号のホップ数をチェックし、最も小さい値を受信した入力ポートに対し、Acknowledgment を送出する。この Acknowledgment は、応答信号がたどった経路を逆方向に転送され、最初に受信したキャッシングノードが Download 元となる。
- 4) Content Download : Request Node ID が与える最短経路を用いてダウンロードされ中継ノードにおいてキャッシュされる。

本アクセスプロトコルの特徴としては、

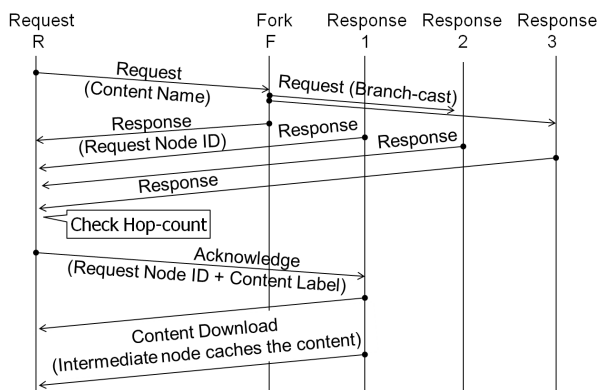
- 1) Content Name は主に Content Request のフェーズで用いられる。
- 2) コンテンツ転送に用いる Content Label は、各ノードがローカルに設定可能である。
- 3) コンテンツの転送制御は、Content Label + Node ID のセットで実現される。という点があげられる。

## 5 計算機シミュレーションによる性能評価

### 5.1 評価項目及び評価対象

本研究の目的は、要求ノードにより近接するキャッシングノードを発見することであり、その有効性を評価するため、発見されたキャッシングノードからのダウンロードホップ数と、探索のために要するオーバーヘッドを評価項目とする。探索オーバーヘッドとしては、要求信号の転送ホップ数及び Neighbor Notification 制御のためのホップ数の和を検討する。

評価する LTH 方式としては、Neighbor Notification 制御を付加しない LTH (S)、LTH (M)、付加する LTH (S)+NN (AL)、LTH (S)+NN (RQ) 及び LTH (M)+NN (AL)、LTH (M)+NN (RQ) の6方式を対象とし、図2の CCN 方式、Flooding 方式及び図6に示す Multiple Breadcrumbs 方式を比較対象とする。図2(b)に示す Breadcrumb 方式では、ノード A のみが Breadcrumb 制御を行っている。これに対して、Multiple Breadcrumbs 方式では、Name base path 上のキャッシュ履歴を有する複数のノード A、B が Breadcrumb 制御を行い、加えて、要求信号は、コンテンツをキャッシュしているノード C (Default Response Node) にも転送される。この方式は通常の Breadcrumb 方式に比べコンテンツ探索能力が強化されている。



(a) Operation Flow

Request Node ID	Content Label	Hop Count	Type	Content Name
-----------------	---------------	-----------	------	--------------

(b) Frame Header Format

図5 アクセスフロー及びフレームヘッダ

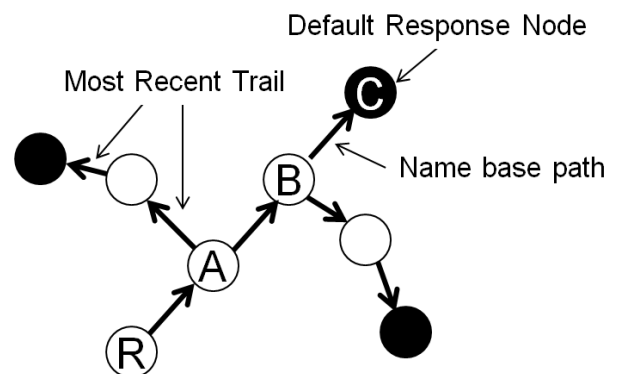


図6 Multiple Breadcrumbs

Mean Request Frequency per Node	10,000 (Total 470,000)
Content Catalog Size	12,800
Buffer Size	32, 64, 128, 256, 512

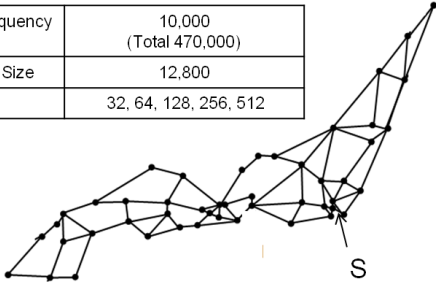


図7 ネットワークモデル及び諸元

## 5.2 シミュレーションモデル

図7にモデルネットワーク及び評価諸元を示す。ネットワークは我が国における光ネットワークの将来像<sup>[3]</sup>として検討されたもので、東京都に2つ、他都道府県には1つのノードが配置される構成である。また、Web ServerはノードS(東京都)に設置するものとする。ノードS以外の47ノードは、それぞれランダムにコンテンツ要求を10,000回行い、12,800種類のコンテンツに対する要求頻度分布は、

$$p(i) = (1/i) / \sum_{k=1}^N (1/k)$$

where  $N = \text{Content Catalog Size}$

で表される Zipf 則 (発生確率は、コンテンツのランキング順位の逆数に比例する) に従うものとする<sup>[4]</sup>。各ノードのバッファサイズは、32、64、128、256、512とし、バッファが満杯になった場合は、Least Recently Used (LRU) 則に従い廃棄 (Eviction) されるものとする。図8は、Zipf 則における累積発生確率を示すもので、第1位ランキングのみで10%、ランキング85位までで50%を占める。

## 5.3 シミュレーション結果

図9に、横軸を各ノードのバッファサイズとするシミュレーション結果を示す。なお、要求ノードがコンテンツを既にキャッシュしている場合は、ダウンロードホップ数や探索ホップ数はゼロとしている。同図(a)に探索されたキャッシュノードと要求ノードとの間のダウンロードホップ数を示す。CCN方式が最も悪く、次いでLTH(M)方式、Multiple Breadcrumbs (MPL-BCR)方式の順で悪い。最近接キャッシングノードを発見する Flooding方式が最も性能が良い。同図(b)は、Flooding方式のダウンロードホップ数に対する各方式の相対値を示す。LTH(M)+NN(AL)方式、LTH(M)+NN(RQ)方式及びLTH(S)方式は、ほぼ同じ性能を実現している。最も性能が良いのは、LTH(S)+NN(AL)方式及びLTH(S)+NN(RQ)方式で、ほぼ同じ性能である。

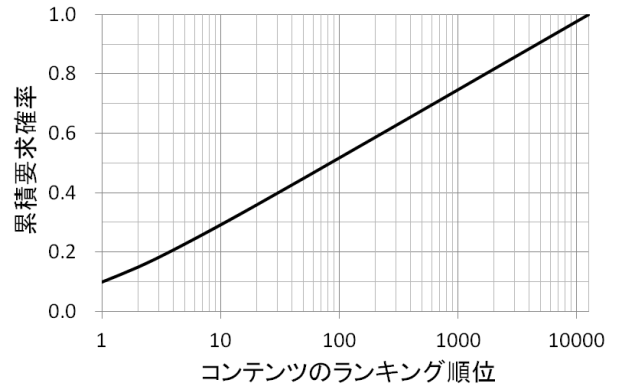
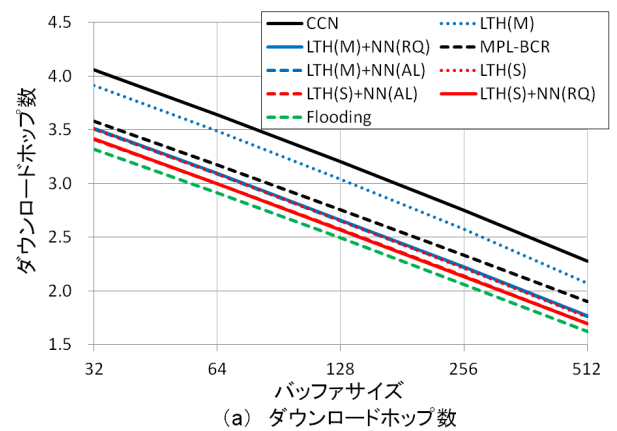
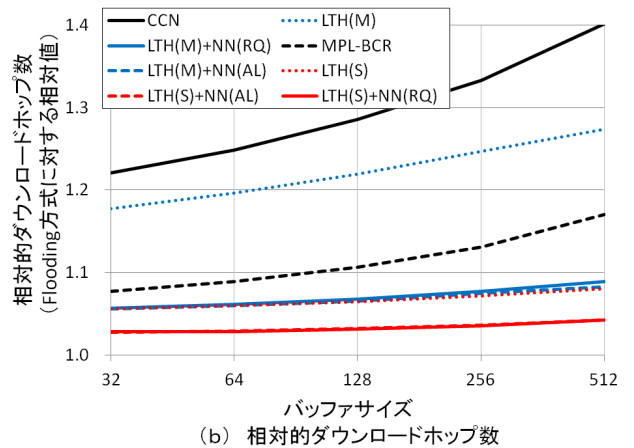


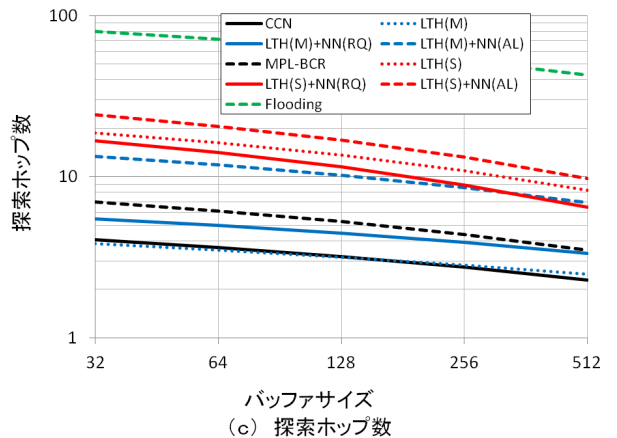
図8 Zipf 則でのコンテンツ要求確率分布



(a) ダウンロードホップ数



(b) 相対的ダウンロードホップ数



(c) 探索ホップ数

図9 性能評価結果

一方、探索のためのオーバーヘッドである、コンテンツ要求の転送ホップ数及び Neighbor Notification のためのホップ数の和で与えられる探索ホップ数の評価結果を同図 (c) に示す。CCN 方式及び LTH (M) 方式が最も小さく、次いで LTH (M) + NN (RQ) 方式、Multiple Breadcrumbs 方式の順で小さい。LTH (M) + NN (RQ) 方式は、Flooding 方式に比べ、探索オーバーヘッドを 1/15-1/10 規模に低減しても、ダウンロードホップ数性能の低下を 5-10% 規模に抑えることができる。LTH (S) + NN (RQ) 方式は、LTH (M) + NN (RQ) 方式との比較で、2.5 倍の探索オーバーヘッドを

費やして、ダウンロードホップ数の改善はわずか 4% 弱である。したがって、探索オーバーヘッドを抑えつつ、近接キャッシングノードを発見する方式としては、LTH (M) + NN (RQ) 方式が望ましい、と言える。

次に、コンテンツのランキング順位に対する性能を評価する。12,800 個のコンテンツを要求確率がそれぞれ 10% となるよう、図 8 に従って 10 グループに分ける。ランキング 1 位のコンテンツは、それだけで 10% を占めるので 1 番目のグループ 1 を構成し、2-4 位、5-11 位、12-31 位、32-85 位、86-231 位、232-631 位、632-1720 位、1721-4693 位で、それぞれ 10% 占有グループ 2、3、... を、4694 位以下で 10 番目のグループ 10 を構成する。

図 10 に、バッファサイズ 128 の場合における性能を示す。同図 (a) に示すように、ダウンロードホップ数の値は、ランキング上位の 3 グループ及び下位の 3 グループにおいては、探索方式による大きな差異が生じていない。コンテンツが広く頒布している場合 (上位) や、逆に Eviction によりキャッシュコンテンツが短時間で廃棄されてしまう状況 (下位) では、方式の違いによる差異が小さいということである。LTH (M) + NN (RQ) 方式と LTH (S) + NN (RQ) 方式との比較で言えば、ダウンロードホップ数は、グループ 5 まですなわちランキング上位 50% のコンテンツに対してはほとんど差がない、と言える。

推奨する LTH (M) + NN (RQ) 方式を、以下にまとめる。

- 1) キャッシュ履歴を有するノードのうち、Name base path 上でコンテンツ名を含むコンテンツ要求信号を最初に受信するノード (Fork Node) は、自身を Root とする Local Tree に対し、要求信号をキャッシュ履歴に基づき Branch-cast する。Branch-cast により要求信号を受信したキャッシングノードは、要求ノードに対し、最短経路を用いて応答信号を送出する。
- 2) Fork Node がキャッシュしていない場合は、1) の処理に加え、要求信号が Name base path 上の最初に受信するキャッシングノード (Default Response Node) まで転送され、該ノードは応答信号を送出する。
- 3) 要求ノードは、コンテンツをダウンロードした時点でキャッシュした旨、全ての隣接ノードに通知する (Neighbor Notification)。
- 4) 1) のフェーズにおいて、コンテンツ要求を受信した中継ノードが、隣接ノードからキャッシュがある旨通知されている場合、この隣接ノードに向けて要求信号を転送 (Short-cut) する。この隣接ノードは、Fork Node として機能する。

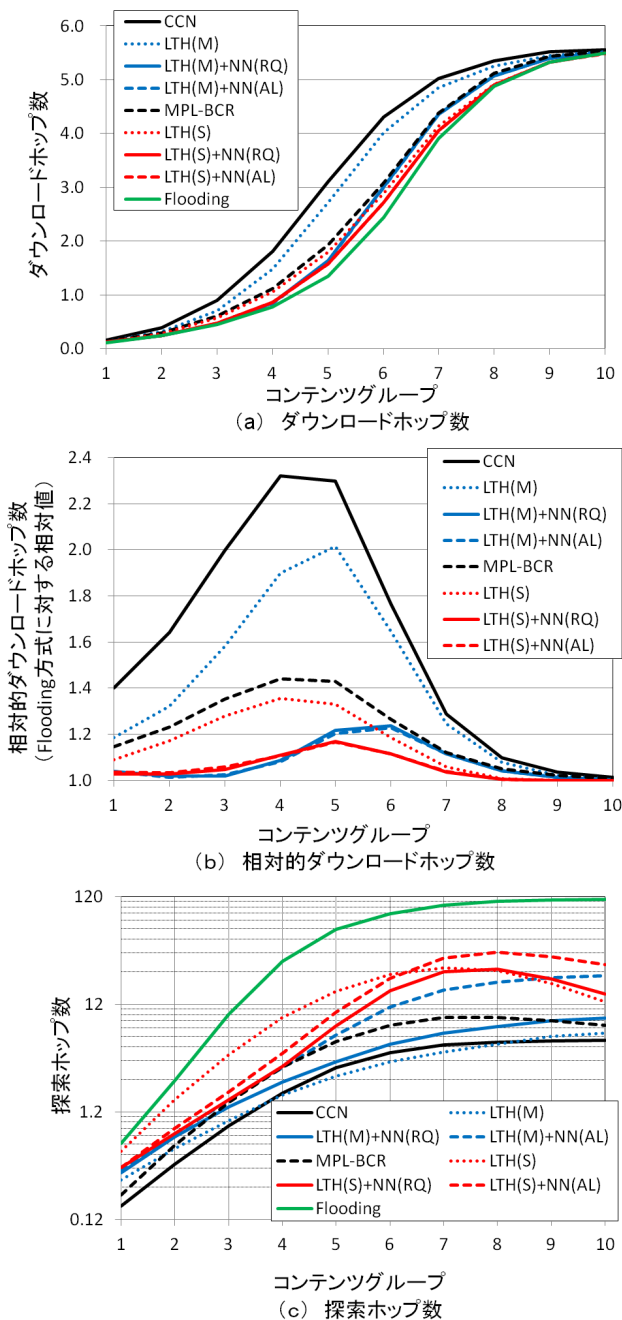


図 10 性能評価結果 (バッファサイズ=128)

- 5) Local Tree 上のキャッシングノード及び Default Response Node からの応答信号を受信した要求ノードは、応答信号の転送ホップ数をチェックし、最も少ない応答ノードに対して Acknowledgment 信号を送出する。
- 6) 応答信号及びコンテンツのダウンロードは、要求ノードへの最短経路を用いるので、Location ID をコンテンツ名に加えて用いる。

## 6 まとめ

コンテンツを網内キャッシュする Information Centric Network において、探索オーバーヘッドを抑えつつ、要求ノードに、より近いキャッシングノードを探索・発見する方法について、幾つかのオプションを含む方式の提案を行い、計算機シミュレーションにより、最も有効な方式の絞り込みを行った。ダウンロード経路を最短化するために信号フレームのヘッダ情報として、要求ノードの Location Address を用いている。

ランキング順位の低いコンテンツは、Eviction によりキャッシュ効果が乏しいので、図 8(c) に示すように、探索オーバーヘッドが大きいという課題が残っている。今後は、ランキング順位(人気度)の低いコンテンツに対し、本課題を解決する改善策につき検討を深める予定である。

## 謝辞

日頃、ご指導いただく大阪大学 村田正幸教授、並びに西永室長はじめネットワークシステム総合研究室の関係各位に深謝します。

### 【参考文献】

- 1 <http://www.named-data.org/ndn-proj.pdf>, "Named Data Networking (NDN) Project," NDN-0001, Oct. 31, 2010.
- 2 Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard, "Networking Named Content," Proc. of CoNEXT 2009, pp.1-12, Dec. 2009.
- 3 Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman, "A Survey of Information-Centric Networking," IEEE Communications Magazine, pp.27-36, July 2012.
- 4 Md. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, Raouf Boutaba, and Bertrand Mathieu, "A Survey of Naming and Routing in Information-Centric Networks," IEEE Communications Magazine, pp.44-53, Dec. 2012.
- 5 H.S. Jeon, I.S. Choi, B.J. Lee, and H.Y. Song, "A Closer Look at Content-Centric Internet Research Projects," ICACT2012, , pp.698-702, Feb. 2012.
- 6 Sen Wang, Jun Bi, Jianping Wu, Zhaogeng Li, Wei Zhang, and Xu Yang, "Could In-Network Caching Benefit Information-Centric Networking?" AINTEC '11 Proc. of the 7th Asian Internet Engineering Conference, pp.112-115, 2011.
- 7 Elisha J. Rosensweig and Jim Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," Proc. of IEEE INFOCOM 2009, pp.2631-2635, 2009.

- 8 H. Shimizu, H. Asaeda, M. Jibiki, and N. Nishinaga, "Content Hunting for In-Network Cache: Design and Performance Analysis," Proc. of IEEE ICC 2014, pp.3178-3183, June 2014.
- 9 H. Shimizu, H. Asaeda, M. Jibiki, and N. Nishinaga, "Local Tree Hunting: Finding Closest Contents from In-Network Cache," IEICE TRANS. INF.& SYST., Vol.E98-D, No.3, pp.557-564, March 2015.
- 10 Lijun Dong, Dan Zhang, Yanyong Zhang, and Dipankar Raychaudhuri, "Optimal Caching with Content Broadcast in Cache-and-Forward Networks," Proc. of IEEE ICC 2011, pp.1-5, June 2011.
- 11 清水洋, 地引昌弘, 朝枝仁, 西永望, "サービス拡張可能な LabelFlow/SDN 及び Information Centric 応用," 電子情報通信学会 信学技報 IN2013-55, pp.113-118, July 2013.
- 12 S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri, "Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed," Computer Networks, Vol.57, Issue 16, pp.3207-3221, Nov. 2013.
- 13 坂野寿和, 築島幸男, 長谷川浩, 釣谷剛宏, 廣田悠介, 荒川伸一, 戸出英樹, "日本の地域特性を考慮したフォトリックネットワークモデルの検討," 電子情報通信学会 信学技報 PN2013-1, pp.1-6, June 2013.
- 14 L. Breslau, Pei Cao, Li Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications," Proc. of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '99, Vol.1, pp.126-134.



清水 洋 (しみず ひろし)

ネットワーク研究本部ネットワークシステム  
総合研究室研究員  
博士(情報学)  
新世代ネットワーク、情報指向ネットワーク、  
SDN



地引昌弘 (じびき まさひろ)

ネットワーク研究本部ネットワークシステム  
総合研究室専門研究員  
博士(システムズ・マネジメント)  
新世代ネットワーク、情報指向ネットワーク、  
超大規模情報流通ネットワーク