

NDNにおけるネットワーク遅延に基づいたキャッシングへのユーザ位置の反映

山口 直樹[†]

[†] 早稲田大学大学院基幹理工研究科 〒169-0072 東京都新宿区大久保 3-14-9

E-mail: †naoki.11_13@asagi.waseda.jp

あらまし IP ネットワークは設計当時現在トラフィックの大部分をコンテンツ要求が占めることが想定されていなかった。そのため、位置指向のネットワークモデルである IP ネットワークに対して、コンテンツ指向のネットワークモデルである情報指向ネットワーク (ICN) が提唱されている。本稿では ICN の実装の 1 つである NDN (Named Data Network) におけるルータでの新しいキャッシュアルゴリズムを提案し、評価を行った。

キーワード ICN, NDN, LRU, キャッシュ手法

Reflecting user location to caching based on network delay in NDN

Naoki YAMAGUCHI[†]

[†] School of Fundamental Science and Engineering, Waseda University Okubo 3-14-9, Shinjuku-ku, Tokyo, 169-0072 Japan

E-mail: †naoki.11_13@asagi.waseda.jp

Abstract At the time of design of the Internet, the content request that currently occupies most of the traffic was not assumed. ICN, which is an information-oriented network model, has been proposed counter to IP networks, which is a position-oriented network model. In this paper, we propose and evaluate a new caching algorithm at intermediate nodes in NDN (Named Data Network), which is one of ICN projects.

Key words ICN, NDN, LRU, Cache Policy

1. ま え が き

インターネットは日々の研究によりめまぐるしい速度で発展してきた。その中で、大型で高価だったコンピュータが、小型で安価になったことにより世間にインターネットによる通信が普及し、インターネットの通信モデルと利用方法との間に齟齬が生まれてきた。この齟齬を解決することでトラフィックの軽減や通信の高速化が期待される。IP アドレスに基づく現在のネットワークはコンテンツ要求がトラフィックの大部分を占める現状に即していない。そのため、IP アドレスに基かない、コンテンツ名に基づいた通信モデルである ICN [4] (Information Centric Network) が提唱されている。ICN にはコンテンツ名に基づいてコンテンツ要求を行い、送られてきたコンテンツを中間ノードでキャッシュするという機能がある。これによって、通信の高速化やトラフィックの軽減、これからの活躍が期待されている IoT への親和性など様々な課題の解決が期待されている。ICN に関わる研究としては、NDN [1] (Named Data Networking), CCN [7] (Content Centric Networking), PURSUIT [2], Net-

Inf [3] などがあり、本稿では ICN の研究の中でも研究が盛んな NDN におけるルータでのキャッシュヒット率を向上させる方法について検討する。

2. Information Centric Networking (ICN)

2.1 ICN とは

ICN はネットワークレイヤでのコンテンツ識別子による通信を実現するプロトコル体系である。つまり ICN は、CDN や P2P のような従来の IP 通信ネットワーク上で成り立つ技術ではなく、ネットワーク基盤を大きく変える通信モデルである。

現在のインターネットでは、ユーザが欲しいコンテンツを取得する時、そのコンテンツの URL 情報を得て、DNS サーバにその URL 情報に対応した IP アドレスを要求し、その IP アドレスの指すサーバからコンテンツを取得している。そのため、IP ネットワークでは IP によるルーティングが行われるため、コンテンツを取得する際にサーバとやり取りする必要があり、これによってオーバーヘッドが生じてしまう。一方、ICN では欲しいコンテンツの名前をもとにコンテンツの要求を行い、

データに直接アクセスすることができる。さらに、ICN ではルータにコンテンツをキャッシュするため、サーバまでアクセスせずにコンテンツを取得できる可能性がある。これにより、ICN ではトラフィックの軽減や通信の高速化が期待される。

ICN の大きな利点は、このように欲しいコンテンツの名前により直接コンテンツを入手することで、コンテンツを取得するためにサーバとやり取りする手間を軽減し、インターネットの通信モデルと使用方法との間の齟齬を解消することができることである。

2.2 ICN の特徴

ICN の特徴を、以下の 4 つの観点から説明する。

- ルーティング

ICN では、IP アドレスではなく、コンテンツ名によってルーティングを行う。コンテンツ名は IP ネットワークにおける IP と違い、コンテンツの位置に依らないため、コンテンツの分散管理がしやすくなる。今後予想されるコンテンツ要求の増加に対し、ICN は効果的である。

- キャッシング

ICN において、Data パケットがホストに返送される時、経路上のルータでコンテンツをキャッシュする。そして、コンテンツを要求する時、要求したコンテンツがルータにあればそのコンテンツを取得する。このとき、要求したコンテンツが存在するノードがユーザに近いほど、ユーザまでのホップ数が少なくなり、トラヒックの削減や通信遅延の短縮に繋がる。

- セキュリティ

ICN ではコンテンツ配送に使う ICN メッセージそのものに認証機能が含まれており、第三者が送信したのコンテンツの信頼性を確かめることができる。

- モビリティ

ICN はコンテンツ要求の際に位置に関わらずコンテンツ名に基づいて通信を行うので、モビリティへの対応が容易である。

3. Named Data Networking(NDN)

3.1 NDN とは

NDN は現在研究されている ICN の仕組みの 1 つである。NDN における通信は Interest パケットと Data パケットという 2 種類のパケットによって実現される。Interest パケットは、ユーザの要求するコンテンツを識別するための名前を記述するフィールドを持ち、ユーザがコンテンツを要求するときにユーザからサーバに向けて送信される。Data パケットは Interest パケットによって要求されたコンテンツを返送するためのパケットで、Interest パケットが通ってきた経路をたどってユーザまで送られる。NDN が扱う Interest パケットと Data パケットの構成を図 1 に示す。

NDN において、各ルータは CS(Content Store)、FIB(Forwarding Information Base)、PIT(Pending Interest Table) という 3 種類のデータ構造を持っている。CS は Data パケットがルータを通過する際にそのコンテンツをキャッシュしておくために用いられる。FIB にはコンテンツ名の prefix とその prefix と一致した Interest パケットの転送先ネットワー

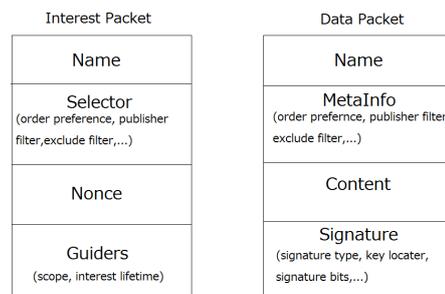


図 1 NDN パケット

クインターフェース (フェイス) が記録されており、Interest パケットの転送方向を示す役割がある。PIT は到着した Interest パケットのコンテンツ名とその Interest パケットの送信元フェイスを記録しておくためのデータ構造である。これは Data パケットを Interest パケットの経路を辿ってユーザまで返送するために利用される。

3.2 NDN におけるコンテンツ取得法

NDN におけるコンテンツ取得法について説明する。ネットワークでユーザがサーバに要求を行うとき、まずユーザはルータに欲しいコンテンツ名を記憶した Interest パケットを送信する。Interest パケットがルータに到着すると、CS を参照し Interest パケットに記憶されたコンテンツ名と一致するデータがキャッシュされていないか確認を行う。一致するデータがキャッシュされていた場合、そのコンテンツを記憶した Data パケットをユーザに返送する。一致するデータが見つからなかった場合は、FIB に記載された情報をもとに次のルータにパケットを送信する。次のルータに Interest パケットを送る際には、返送された Data パケットが Interest パケットの経路を辿る際に必要になる、Interest パケットに記されたコンテンツ名と、Interest がどのフェイスから来たのかという情報を PIT エントリとして、PIT に保存しておく。そしてこの手順を次のルータでも同様に繰り返す。途中のどのルータの CS にもコンテンツがキャッシュされていなかった場合は Interest パケットがサーバまで辿り着くため、そこでコンテンツを取得する。要求されたコンテンツを含む Data パケットは PIT に記録されている情報をもとに、コンテンツを要求したユーザに届けられる。このように、NDN ではネットワーク内の複数のルータにコンテンツをキャッシュしておくことで、目的のコンテンツが中間ルータにキャッシュしてあれば、効率よくコンテンツを取得することができる。このことから、中間ルータにおけるキャッシュヒット率が高くなることが望まれる。そのため、本稿では NDN におけるキャッシュ方式の提案を行う。

3.3 キャッシュ判断方式

この節では、Data パケットが中間ルータに到着したときそのコンテンツをキャッシュするか否かを判断するアルゴリズムについて説明する。適切な位置にキャッシュを行うことで、キャッシュヒット率を向上し、より短い時間でコンテンツの取得を行うことができる。ここでは、確率的キャッシュ [6] について説明する。コンテンツ取得のため Interest パケットがサー

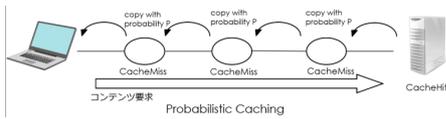


図2 確率的キャッシュアルゴリズムのオペレーション

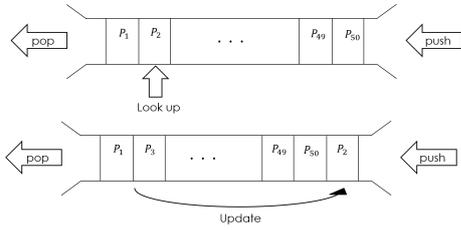


図3 LRU アルゴリズム

バに向けて送信され、コンテンツがヒットした場所までの経路上のすべてのルータで確率 P を用いてランダムにキャッシュ手法を使用した。サーバから遠い位置にあるルータでのキャッシュ確率 P を高め、近い位置にあるルータではキャッシュ確率 P を低くする。このとき、キャッシュのための重みは、ルータごとに、ユーザからサーバまでのホップ数と、キャッシュするルータからサーバまでのホップ数をもとに計算する。無駄なキャッシュを減らし、より良いキャッシュを実現できるという利点がある。

3.4 キャッシュ置換方式

パケットを中間ルータにキャッシュする時にキャッシュメモリがいっぱいだった場合、どのコンテンツを廃棄して新しいコンテンツをキャッシュするかを決定するのがキャッシュ置換方式である。適切なコンテンツを廃棄することで、キャッシュサイズを増やすことなくキャッシュヒット率を上昇させることができる。より適切なコンテンツを廃棄するためには、コンテンツ毎の最後にアクセスされた日時や一定期間で何度アクセスされたかといった使用情報を十分に参照する必要がある。一方で、使用情報が多くなると、情報を更新するためのコストが大きくなり、データが要求されてから、それを返すまでの遅延時間も大きくなる。ここでは本稿で扱う LRU について説明する。最後に利用されてからの経過時間が最も長いデータを最初に捨てるという手法である。本稿で扱うシミュレータ ndnSIM では双方向 list クラスを用いて、データが参照されたときにそのデータを list の最後尾に移動させることで、この手法を実現している。この手法の様子を図3に示す。

4. 提案手法

ICN の初期設計では CEE というヒットした場所からユーザまでの経路上のすべてのルータにおいて、送られてきた Data パケットのコンテンツをキャッシュする手法が考えられてきた。しかし、CEE は同一経路上に同じコンテンツが複数キャッシュされてしまうため、メモリ資源の無駄が多いなど欠点が多く、キャッシュヒット率も低くなってしまった。そこで、本稿ではネットワーク遅延を考慮した重み [6] における、キャッシュを行う中間ノードとユーザの位置関係を考慮した LRU [5] の有

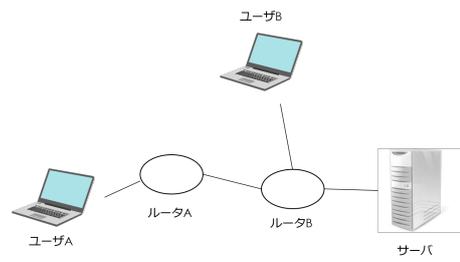


図4 LRU の課題点

用性を検証する。

4.1 提案における背景

確率的キャッシュの操作は 1) 確率 P を定めキャッシュするかどうかを判断する。2) キャッシュする場合キャッシュの一番後ろにコンテンツを挿入する。といった2段階に分割して考えられる。1) では2つの問題点が挙げられる。1つ目は従来の確率的キャッシュでは「ユーザからサーバまでの hop 数」と「キャッシュするルータからサーバまで hop 数」から重みを計算している点である。実際のネットワークでは同じ hop 数でもネットワーク遅延が異なる場合がある。そのため、実際のネットワークでは最適なキャッシュが行えない恐れがある。もうひとつは、キャッシュが埋まるのに時間がかかる点である。確率的キャッシュでは確率に基づいてキャッシュを行うかどうか決めるため、各ルータでのキャッシュ回数は CEE に比べて少なくなり、結果としてキャッシュが最適化されるまでに時間がかかる。

2) でも問題点が挙げられる。この問題点について、図4を用いて説明する。1つはサーバから離れたユーザの影響が大きいという点である。図4においてユーザAがコンテンツ要求を行う時、ルータA、Bにおいてキャッシュが行われる。一方で、ユーザBがコンテンツ要求を行ったときは、ルータBにしかキャッシュが反映されない。つまり、ユーザBは、直近に存在するルータBにしかコンテンツをキャッシュすることができない。もう1つの問題点はサーバに近いルータに、人気のないコンテンツが集中してしまうという点である。図4においてユーザAが人気なコンテンツを要求した時、ルータAでコンテンツがヒットし、サーバの近くを Interest パケットが通過しない。これによって、ルータBにはほかのルータにはキャッシュされていない人気のないコンテンツがキャッシュされてしまう。

これらの問題の影響を抑えるために、本稿では hop 数ではなくネットワーク遅延から重みを計算し、重みに従った生存時間を設定して 100%キャッシュを行う LRU 方式を採用することを提案する。

4.2 提案手法

現在提案されている NDN のルータにおけるキャッシュ制御手法には、4.1 で述べたとおり、4つの問題点が挙げられる。これらの問題を軽減する手法を提案する。提案手法の実行手順を次に示す。Data パケットがルータに到着すると、そのルータの位置に従った重み P を計算する。CS の中に同じコンテンツがキャッシュされている場合、重みに従った位置にキャッシュ

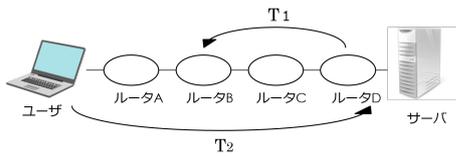


図5 ネットワーク遅延を用いた重みの取り方

を更新する。CSの中に同じコンテンツがキャッシュされていない場合、重みに従った位置に新しくキャッシュを挿入する。これによって、Data パケットがルータを通過するとき重みに従った新しさとしてキャッシュを保持することができる。提案手法の実装方法について以降の節で詳しく述べていく。

4.3 ネットワーク遅延への置き換え

NDNの実装を考えた時、実際のネットワークでは同じホップ数でもネットワーク遅延が異なる場合がある。このとき、ネットワーク遅延が大きいルータより、ネットワーク遅延が小さいルータの方が応答時間が短く、距離が短いと言える。そこで、各ルータの重みを計算する際に hop 数でなく、ネットワーク遅延を用いることで、現実に即した重みを計算する。例えば、2ホップの距離にあるネットワーク遅延の小さい場所にあるルータと1ホップの距離にあるネットワーク遅延の大きい場所にあるルータでは、2ホップの距離にあるルータの方を高い確率でキャッシュする。このときの重みの取得法を図5を用いて説明する。あるルータにData パケットが到着したとき、ユーザからサーバまでのネットワーク遅延を T_2 、サーバからキャッシュするルータまでのネットワーク遅延を T_1 とおく。このとき、そのルータの重み P を

$$P = T_1/T_2 \quad (1)$$

と定義する。重み P は、ユーザに近いルータほど大きくなる。ユーザに近い場所ではキャッシュの生存時間を長くし、遠い場所では生存時間を短くすることで、無駄なキャッシュを少なくし、効率の良いキャッシュを行うことができる。

4.4 ネットワーク遅延時間の取得方法

重みを求めるために必要なネットワーク遅延 T_1 、 T_2 の取得方法を説明する。

• T_1 の取得方法

図6において、ルータBに注目して T_1 の取得法を説明する。まず、ユーザがサーバに向けて Interest パケットを送信する。そして、各ルータに Interest パケットが到着したとき、PITにコンテンツ名とフェイスの情報の他に、到着した時刻を併せて記録する。ルータBに到着したときの時刻を t_1 とする。その後、Interest パケットがサーバまで到達し、ユーザに向けてコンテンツを乗せた Data パケットが送信される。Data パケットはPITを参照しながらユーザの元まで運ばれる。このとき、Data パケットが各ルータに到着した時間をそれぞれPITに記載する。Data パケットがルータBに到着したときの時刻を t_2 とする。この2つの時刻 t_1 、 t_2 から、サーバからキャッシュするルータBまでのネットワーク遅延 T_1 を、以下の式で求める事ができる。

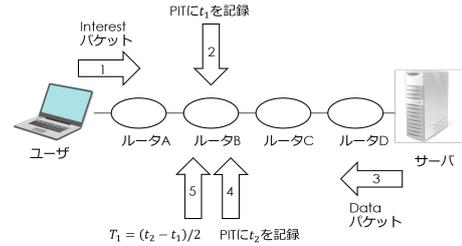


図6 T_1 の取得方法

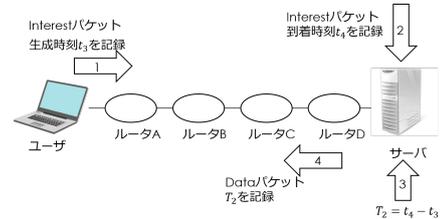


図7 T_2 の取得方法

$$T_1 = (t_2 - t_1)/2 \quad (2)$$

• T_2 の取得方法

図7に T_2 取得の様子を示す。まず、ユーザがサーバに向けて Interest パケットを送信する。この時、Interest パケットが生成された時刻を Interest パケット自身に記録する。このときの時刻を t_3 とする。Interest パケットがサーバまで到達し、ユーザに向けてコンテンツを乗せた Data パケットが送信される。このときの時刻を t_4 とする。この2つの時刻 t_3 、 t_4 から、ユーザからサーバまでのネットワーク遅延 T_2 を以下の式から求められる。

$$T_2 = t_4 - t_3 \quad (3)$$

4.5 キャッシュ位置

本稿では、ユーザ及びキャッシュを行うルータとサーバの位置関係をキャッシュの生存時間に反映させたLRUを採用する[5]。具体的には、ユーザ位置を示す重み P を利用してキャッシュの新しさを変化させることで、生存時間の制御を行う。この手法の動作について図8を用いて説明する。まず、ユーザがサーバに向かってコンテンツ要求のための Interest パケットを送信する。そして、Interest パケットがサーバに到達するとそのコンテンツが保持された Data パケットが生成され、ユーザに向かって返送される。このとき、4.3節で述べたとおり、各ルータでは重み P が算出される。図8に記載されている確率は P の例である。従来のLRU、確率的キャッシュでは、最も新しいコンテンツとしてキャッシュの末尾に挿入を行っていたが、この手法では、重みに従った新しさの位置にキャッシュの挿入を行う。図9では図8と同じトポロジでキャッシュが行われるときにキャッシュがどこに挿入されるかを示す。i番目のルータ i がもつキャッシュ容量を N_i とし、右にあるほど新しいコンテンツで生存時間が長く、左にあるほど古いコンテンツとして早く廃棄される。図9ではユーザに隣接するルータAでは重みが100%のためキャッシュを行う場合はもっとも新しいキャッシュとしてCSの末尾にキャッシュする。一方、ルータ

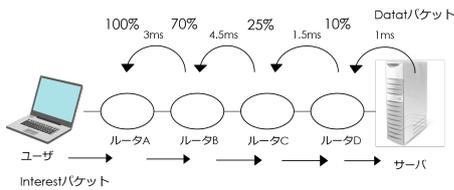


図 8 ネットワーク遅延を用いた重みの例

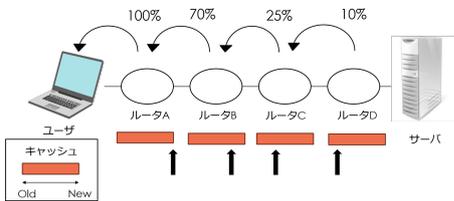


図 9 提案手法の動作

Cでは重みが25%のため、CSの先頭から25%の位置にキャッシュする。他のルータでも同様の処理を行う。

Interest パケットが要求したコンテンツが、もし中間ノードでヒットした場合には、ルータに保持されていたキャッシュを、キャッシュしたときの重みを参照して、その重みの分だけキャッシュの新鮮さを更新する。

4.6 提案手法の利点

提案手法についてまとめる。Data パケットがルータに到着すると、まず 4.3 節で説明したように、ルータの重みを計算する。そして、4.5 節で述べたように CS 中の重みに従った位置にキャッシュを行う。この手法には、確率的キャッシュや LRU には見られなかった 2 つの利点がある。この 2 つの利点について説明する。

- ネットワーク遅延に基づいた重み

図 10 を用いて説明する。ユーザ A とルータ A、ユーザ B とルータ C の間のネットワーク遅延及びユーザ A とサーバ A、ユーザ B とサーバ B のネットワーク遅延はそれぞれ等しいとする。このとき、hop 数で重みを計算した場合、サーバ A に隣接するサーバまでのネットワーク遅延が 10ms のルータ B と同じくサーバ B に隣接するネットワーク遅延が 20ms のルータ D を同じ重みでキャッシュするが、ネットワーク遅延に基づく重みを計算した場合、よりユーザに近いルータ D ではルータ C より生存時間を長く取ることができる。

- ルータとユーザとの距離による使用可能なキャッシュ領域の制御

図 11 においてユーザ A がコンテンツ要求をサーバに対して行ったとき、ルータ A での重みを 100%、ルータ B での重みを 50% とすると、ルータ A では一番新しい位置に、ルータ B では 50% の新鮮さの位置にキャッシュを行う。一方、ユーザ B がコンテンツ要求を行うとき、ルータ B の一番新しい位置にキャッシュを行うことができる。これによって、従来のキャッシュ方式ではユーザ B の要求するコンテンツがすぐに廃棄されてしまっていたが、ユーザ B の要求に対するコンテンツを保持しやすくなる。これによって、遠い位置にいるユーザの影響を軽減

することができる。

- 100% キャッシュによるより速いキャッシュ最適化

図 12 を用いて説明する。重み $P = 100\%$ のとき、 $P = 33\%$ の場合に、通常の LRU と提案手法について考える。キャッシュ容量をコンテンツ 3 つ分とし、コンテンツが初めてキャッシュに到着したときにキャッシュを行い、それ以外の時にそのキャッシュの位置が更新されないと仮定する。また、キャッシュが適正な生存時間をもつときのキャッシュ位置がルータの容量 $N_i \times$ 重み P で表されるとする。このとき、通常の確率的キャッシュでは重み $P = 100\%$ のときは最初の到着の時にキャッシュは適正な生存時間となるが、重み $P = 33\%$ のときは 3 回キャッシュが到着したときようやく適正な生存時間になる。一方で提案手法では、最初の到着の時点で適正な生存時間となる。

5. 提案手法の評価

本章では NDN のシミュレータである ndnSIM を用いてシミュレーションを行い、CEE、ホップ数に基づいた確率的キャッシュ、ネットワーク遅延を考慮した確率的キャッシュの 3 つの手法と提案手法の比較を行った。比較は各ルータにおけるキャッシュヒット率 H によって行う。

$$H = \frac{\text{キャッシュヒット数}}{\text{キャッシュヒット数} + \text{キャッシュミスヒット数}} \quad (4)$$

式 4 におけるキャッシュヒット数は、Interest パケットが

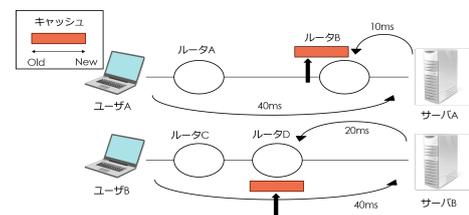


図 10 hop 数とネットワーク遅延の置き換えによる利点

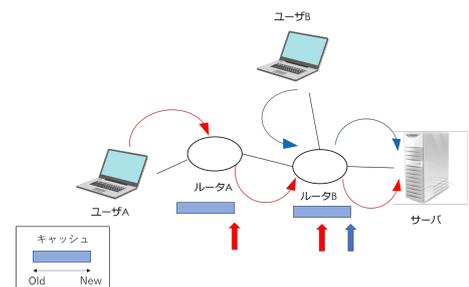


図 11 ルータに隣接するユーザ専用のキャッシュ領域

		1回目	2回目	3回目
ProbCache	P=100%			
	P=33%			
提案手法	P=100%			
	P=33%			

図 12 キャッシュの最適化の様子

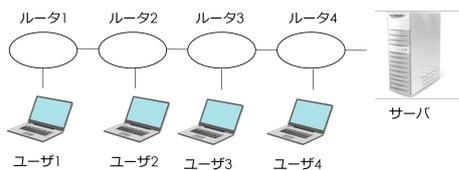


図 13 シミュレーションのトポロジ

表 1 シミュレーションパラメータ

シミュレータ	ndnSIM
トポロジ	ライン型
サーバ数	1
ユーザ数	4
ルータ数	4
ネットワーク遅延	10Mbps/1ms
コンテンツ数	200
キャッシュ容量	100
コンテンツ要求	Zipf の法則

ルータに到着したとき、そのルータに要求されたコンテンツが保持されていた回数を示す。一方、キャッシュミスヒット数は、Interest パケットがルータに到着したとき、そのルータに要求されたコンテンツが保持されていなかった回数を示す。

5.1 シミュレーション

シミュレーションで用いたネットワークトポロジとシミュレーションパラメータを図 13 と表 1 にそれぞれ示す。コンテンツ要求は Zipf の法則に従って行う。Zipf の法則は web サイトにおいてアクセス数が k 番目に多いサイトの全体に占める割合が $\frac{1}{k}$ に比例するという経験則である。サーバから一直線に伸びたトポロジを使用し、ルータが 4 つ配置してある。各ルータにはユーザが一人接続され、それぞれ異なるコンテンツ要求を行う。CEE, ホップ数に基づいた確率的キャッシュ (Probabilistic), ネットワーク遅延を考慮した確率的キャッシュ (ND-Prob) の 3 つの手法と提案手法の比較を行う。計測時間を 20 秒とし、各ルータにおけるキャッシュヒット率を計測した。

シミュレーションによる検証結果を表すグラフを図 14 に示す。提案手法と各手法との比較を行っていく。提案手法と Cache Everything Everywhere(CEE), ホップ数に基づいた確率的キャッシュ (Probabilistic), ネットワーク遅延に基づいた確率的キャッシュ (ND-Prob) を比較したとき、提案手法により全てのルータにおいてキャッシュヒット率が向上したことがわかる。ネットワーク全体でのキャッシュヒット率は CEE が 2.5%, Probabilistic が 2.4%, ND-Prob が 3.4% で、提案手法が 10.4% であった。キャッシュヒット率は、CEE と比較すると 7.9%, Probabilistic と比較すると 8.0%, ND-Prob と比較すると 7.0% 上昇した。ルータ 1 以外のルータではキャッシュヒット率が他の手法と比較して大きく上昇していることが確認できる。これは、従来の手法では、ユーザとルータの位置関係に関わらずキャッシュしていたのを、4.6 節で述べた通り、ユーザとルータの位置関係によって生存時間を調節したことで、離れた位置にいるユーザの影響の軽減ができたためと考えられる。

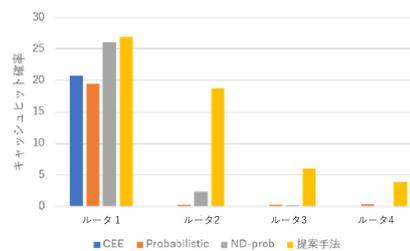


図 14 シミュレーション結果

6. む す び

ネットワーク遅延に基づいたキャッシュ判断アルゴリズムにおける重みを利用してユーザ位置を生存時間に反映し、100%の確率でキャッシュを行うことで、より速いキャッシュの最適化及びキャッシュ領域の使用の効率化を図った。また、ndnSIM によるシミュレーション評価により提案手法の有効性を確認した。今後の課題として、今回の研究では大きなトポロジでの検証を行うことができなかったため、課題として挙げられる。

文 献

- [1] Named data networking: Motivation & details - named data networking (ndn). <https://named-data.net/project/archoverview/>. (Accessed on 02/02/2019).
- [2] Publish subscribe internet technology | projects | fp7-ict | cordis | european commission. <https://cordis.europa.eu/project/rcn/95665/factsheet/en>. (Accessed on 02/02/2019).
- [3] Christian Dannewitz, Dirk Kutscher, B6Rje Ohlman, Stephen Farrell, Bengt Ahlgren, and Holger Karl. Network of information (netinf)—an information-centric networking architecture. *Computer Communications*, Vol. 36, No. 7, pp. 721–735, 2013.
- [4] 伊藤章, 福田健一. Icn が切り開く次世代ネットワークアーキテクチャー, 2015.
- [5] 宮崎貴博, 竹下秀俊, 岡本聡, 山中直明. コンテンツセントリックネットワークにおけるユーザ位置を生存時間に反映したキャッシュ手法の提案. 電子情報通信学会技術研究報告= IEICE technical report: 信学技報, Vol. 114, No. 109, pp. 7–12, 2014.
- [6] 金田健吾, 中里秀則. ネットワーク遅延に基づくキャッシュ方式.
- [7] 山本幹. コンテンツオリエンテッドネットワーク. 信学誌, Vol. 95, No. 4, pp. 341–346, 2012.