

NDNにおけるネットワーク遅延とキャッシュ容量に基づくキャッシュ方式

伊藤 快[†] 中里 秀則[†]

[†] 早稲田大学基幹理工学部情報通信学科 〒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 ICN(Information Centric Networking)の研究の中にあるNDN(Named Data Networking)においてルータのネットワーク遅延とキャッシュ容量に応じてコンテンツをキャッシュする確率を変えることでキャッシュヒット率が向上するかどうかを検証する。

キーワード NDN, キャッシュ方式

Caching method based on network delay and cache capacity in NDN

Konomu ITO[†] and Hidenori NAKAZATO[†]

[†] Faculty of Engineering, Waseda University Okubo 3-4-1, Shinjyuku-ku, Tokyo, 169-8555 Japan
E-mail: †konokono1117@fuji.waseda.jp, nakazato@waseda.jp

Abstract In NDN(Named Data Networking) in ICN(Information Centric Networking) research, I will verify whether changing the probability of content caching according to router network delay and cache capacity will improve the cache hit rate.

Key words NDN, Cache-method

1. ま え が き

現在のインターネット通信はコンテンツを保有する場所をあらわすIPアドレスを用いてhost-to-host間での通信を行うIP通信モデルとなっている。だが、実際のインターネットでは流れるトラフィックの大部分をコンテンツ要求が占めている。この通信モデルとサービスとの乖離をなくすためにICN(Internet Centric Networking)という情報指向ネットワークに関する研究が進められている[7]。本研究ではICNの研究の中でも特に活発に研究が行われているNDN(Named Data Networking)[5]においてコンテンツをより効率的に取得するためにネットワークの遅延時間とルータのキャッシュ容量に基づくキャッシュ方式の検証を行う。

2. NDNの説明

2.1 NDNとは

NDNとはICNの中でも特に活発に研究が行われているプロトコルである。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つのパケットをユーザとルータもしくは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、ユーザが要求するコンテンツ名が記憶されている。

のコンテンツ名に基づいて、データが存在するサーバに向かって送信される。Dataパケットはサーバやルータで取得したコンテンツを運ぶために用いられ、その中にコンテンツが記憶されている。

NDNにおいてルータには、ルーティングやキャッシュデータの取得のためにPIT(Pending Interest Tables), CS(Content Store), FIB(Forwarding Information Base)という3つのテーブルが存在する。

PIT: 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テーブル

CS: ルータでキャッシュされたデータを格納、そのコンテンツ名を記憶するテーブル

FIB: CSとPITに要求されたコンテンツが存在しない場合の次の転送先を示すテーブル

NDNでは上記のPIT, CS, FIBを用いてユーザは要求するコンテンツを取得する。

2.2 NDNにおけるコンテンツ取得方法

NDNにおけるコンテンツの取得方法を図1を用いて説明する。

ユーザは取得したいコンテンツ名を持ったInterestパケット

をデータサーバに向けて、ルータ A, B, C の順に経由して送信する。まず始めにルータ A の CS がユーザの取得したいコンテンツ名のデータを記憶しているかを確認する。もし、ルータ A にユーザが取得したいコンテンツが記憶されていた場合、その後のルータやサーバに Interest パケットを送ることなくルータ A からユーザに向けてコンテンツを記憶した Data パケットを送信する。記憶されていなかった場合は、FIB に従って Interest パケットを次のルータ B に転送する。次のルータに Interest パケットを送信する際、Interest パケットがどこから来たかを PIT エントリとして PIT に記憶する。この操作をルータ A 以降のルータ B, C でも同様に行う。すべてのルータの CS で取得したいコンテンツを記憶していなかった場合は、Interest パケットはサーバに到着する。その際、サーバで取得したいコンテンツ名の Data パケットを各ルータの PIT に記憶されている経路を頼りにユーザに送信する。

このように、NDN ではルータでデータを記憶することができ、そこにユーザが要求するコンテンツ名のデータが記憶されていればルータからユーザにデータを送れるため、サーバからデータを取得するよりもより短時間で効率よくコンテンツを取得できる。このような特徴から、NDN ではユーザとサーバの経路上にあるルータでのキャッシュヒット率が高くなることが望ましい。従って、本稿では NDN 内のルータにおいて効率の良いキャッシュ方式の提案を行う。

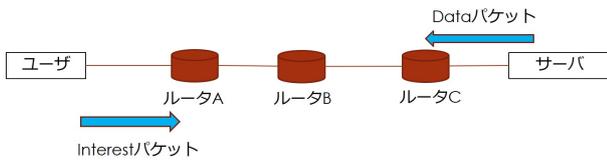


図 1 NDN におけるコンテンツの取得方法

3. NDN におけるキャッシュ手法

本章では NDN におけるキャッシュ手法の研究のいくつかを紹介する。まずキャッシュ手法は大きく以下の 2 つにわけることができる。

- On-path Caching
- Off-path Caching

3.1 On-path Caching

On-path Caching [3] とは、コンテンツをキャッシュするルータが、Data パケットの経路上であるキャッシュ手法である。本稿の提案手法もこの On-path Caching である。On-path Caching を行うキャッシュ手法には Probabilistic Caching, Leave Copy Down [4] といったものがあげられる。

Probabilistic Caching とは、各ルータにおいて、確率 P でコンテンツをキャッシュするかを決める手法である。

Probabilistic Caching は、大きく以下の 2 つに分類できる。Random Caching

Random Caching とは、Interest パケットがサーバに送信さ

れる過程で、辿ってきた経路上のルータの 1 つを、アルゴリズムによって決定されたキャッシュ確率 P でコンテンツをキャッシュするか決める手法である。この手法はとてもシンプルで、ネットワークに負荷がかからないというメリットがある。しかし、デメリットとしてコンテンツの冗長性を高めるといふ点や、キャッシュを行わない可能性があるという点がある。Random Caching の代表的なものが Caching Everything-Everywhere(CEE) だ。これはヒットした場所からユーザまでの経路上に存在するすべてのルータで必ずキャッシュする手法である。

ProbCache (Probabilistic In-Networking Caching) [6]

ProbCache とはサーバから遠い位置にあるルータにおいてキャッシュする確率を高め、サーバから近い位置にあるルータではキャッシュする確率を低くすることで、無駄なキャッシュを少なくし、全体的なキャッシュ確率を上げる手法 [6] である。サーバからの距離でキャッシュする確率を変えるため、ユーザからサーバまでのホップ数を x 、キャッシュするルータからサーバまでのホップ数を y と置いて以下の式よりキャッシュの重み CacheWeight を計算する。

$$\text{CacheWeight} = \frac{y}{x} \quad (1)$$

次に、Interest パケットがサーバに送信される過程で、辿ってきた経路上でキャッシュするコンテンツの数を示す TimesIn を計算する。あるルータ i のキャッシュ容量を N_i とすると以下の式より求めることができる。

$$\text{TimesIn} = \frac{\sum_{i=1}^{x-(y-1)} N_i}{T_{tw} \times N_i} \quad (2)$$

ここで $\frac{\sum_{i=1}^{x-(y-1)} N_i}{T_{tw} \times N_i}$ はユーザからサーバまでの距離が遠いほどキャッシュする重みを増やしていくための値である。 T_{tw} とは、ある一定確率でキャッシュを行うための値であるタイムズウィンドウ [6] を示す。上で示した CacheWeight と TimesIn を用いて ProbCache におけるキャッシュの確率を以下の式で求める。

$$\text{ProbCache} = \text{CacheWeight} \times \text{TimesIn} \quad (3)$$

ProbCache はこのように求めることで、サーバとの距離に応じて確率的にキャッシュし、キャッシュヒット率を上げることができる。本稿ではこのキャッシュ手法と提案手法を比較することで検証を行う。

3.2 Off-path Caching

Off-path Caching とは On-path Caching とは逆で、Interest パケットがサーバに送信される過程で、辿ってきた経路以外の経路でキャッシュが行われる手法である。メリットとして、コンテンツを持っているルータが移動した際、移動前と同様のコンテンツ名でコンテンツを取得できることが容易になるという点がある。デメリットとして、追加のオーバーヘッドがかかってしまう点がある。

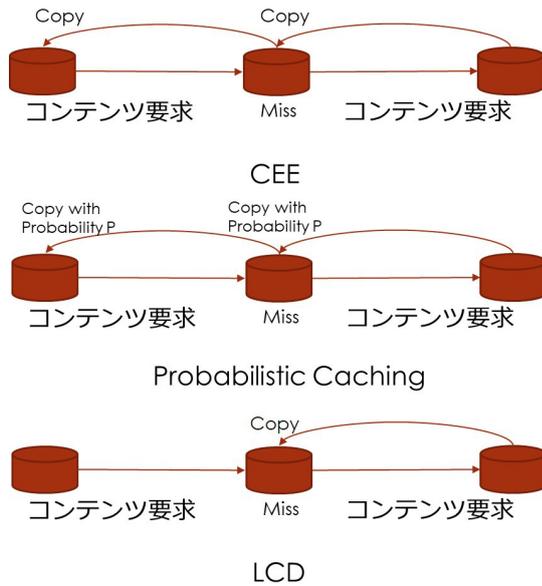


図2 CEE、Probabilistic Caching、LCDの様子

4. 提案手法

NDNにおけるデフォルトのキャッシュ判断方式はCaching Everything-Everywhere (CEE) というものである。

だが、CEEでは経路上に同じコンテンツがキャッシュされてしまうなどの問題点がある。このような問題点により、キャッシュヒット率が低くなってしまい、キャッシュ効率が悪くなっていると考えられる。そこで本稿では、[8]、[9]が行った研究を参考にし、[8]が行った提案手法を[9]の研究で用いたトポロジーでProbCacheと比較し評価する。なお、[8]の研究ではtreeトポロジーを用いて研究を行っていた。

4.1 レスポンスタイムの取得方法

本研究ではネットワーク遅延を考慮し確率計算を行う。そのために必要なパラメータである T_1 、 T_2 の取得方法について説明する。

4.1.1 T_1 の取得方法

図3を用いてルータBで T_1 を取得する方法を説明する。

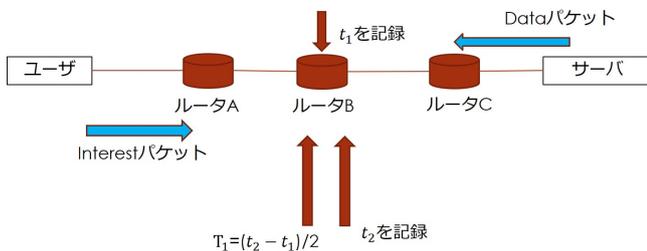


図3 T_1 の取得方法

- (1) ユーザが取得したいコンテンツ名のInterestパケットをルータを経由し、サーバに向けて送信する。
- (2) 送信したInterestパケットがルータBに到着した時刻を t_1 として記録する。
- (3) Interestパケットがサーバに到着し、コンテンツが保

持されたDataパケットをユーザに向けて送信する。

(4) サーバが送信したDataパケットがルータBに到着した時刻を t_2 として記録する。

(5) 2と4で記録した時刻 t_1 、 t_2 を用いて、以下の式でサーバからルータBまでのレスポンスタイム T_1 を取得する。

$$T_1 = \frac{t_1 - t_2}{2} \quad (4)$$

4.1.2 T_2 の取得方法

図4を用いてルータBで T_2 を取得する方法を説明する。

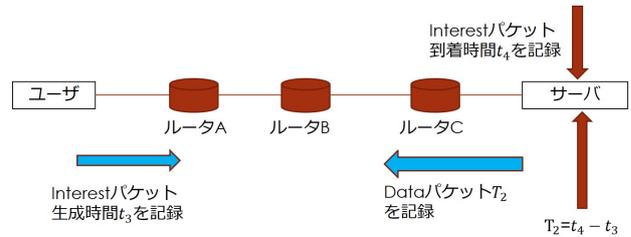


図4 T_2 の取得方法

(1) ユーザが取得したいコンテンツ名のInterestパケットをサーバに向けて送信する際、Interestパケットが生成された時刻を t_3 としてInterestパケットに記録する。

(2) Interestパケットがルータを経由してサーバに到着した時間を t_4 としてInterestパケットに記録する。

(3) (1)と(2)で記録した時刻 t_3 、 t_4 を用いて、以下の式でユーザからサーバまでのレスポンスタイム T_2 を取得する。

$$T_2 = t_4 - t_3 \quad (5)$$

4.2 キャッシュ確率

提案するキャッシュ手法は、以下の式で計算される P の確率で、各ルータを通過するコンテンツのキャッシュを確率的に行うというものである。

$$P_1 = \frac{T_1}{T_2}$$

$$P_2 = \frac{\sum_{i=1}^{x-(y-1)} N_i}{x \times N_{ave}}$$

$$P = P_1 \times P_2 \quad (6)$$

ここで、 T_1 はサーバからキャッシュするルータまでのパケット転送に要する時間、 T_2 はユーザからサーバまでパケット転送するのに要する時間、 x はユーザからサーバからサーバまでのホップ数、 y はキャッシュするルータからサーバまでのホップ数、 N_{ave} はネットワーク全体でのルータのキャッシュ容量の平均を表している。 P_2 はユーザからサーバまでのルータが持つキャッシュ量と、あるルータからサーバまでのキャッシュ量の割合である。

この手法により、ネットワーク遅延とキャッシュ容量を考慮することでネットワーク遅延が大きく、キャッシュ容量が大きいほどキャッシュ確率が高くなる。本稿ではこのキャッシュ確率を用いることで、効率の良いキャッシングになると考える。

5. 評価

本稿では提案手法を NDN のシミュレータである ndnSIM を用いて評価した。比較するアルゴリズムとしては ProbCache (Probabilistic In-Network Caching) を用いた。提案手法と ProbCache のキャッシュヒット率を比較することで評価を行った。キャッシュヒット率は以下の式で定義した。

$$\text{キャッシュヒット率} = \frac{\text{キャッシュヒット数}}{\text{キャッシュヒット数} + \text{キャッシュミスヒット数}} \quad (7)$$

ここで、キャッシュヒット数とはユーザが取得したいコンテンツ名の Interest パケットが途中のルータでコンテンツが保持されていた場合の数を示す。キャッシュミスヒット数とはユーザからサーバまでのルータでユーザが取得したいコンテンツの Interest パケットが保持されていない場合の数を示す。

また提案手法がネットワーク遅延を考慮していることから遅延時間の大小によってもキャッシュヒット率に変化が起きるだろうと考え本稿においてはルータ間のパケット転送遅延時間の幅が小さいときと大きいときの 2 つの場合でも比較を行った。具体的に遅延時間が小さいときは 1ms-5ms、大きいときは 1ms-10ms の中からランダムで各ルータ間の遅延時間を定めた。

ネットワークとして、本研究では実際のトポロジーを使用するため、[9] の研究を参考に、評価に用いるネットワークトポロジーとしては Sinet と Geant の二つのトポロジーを用いた。Sinet [2] とは学術情報ネットワークの別名で、国立情報学研究所 (NII) が構築し運用している、学術利用のための通信ネットワークである。また、Geant [1] とは欧州の研究開発、学術利用のためのネットワークとして、EU と周辺の国々が共同で出資して構築したネットワークである。この 2 つのトポロジーを作成し、シミュレーションを行った。

5.1 シミュレーション

Sinet のトポロジーを図 5 に Geant のトポロジー図 6 を示す。青丸で示しているのがルータであり、赤の線はルータ間の隣接を表している。各ルータにそれぞれ 1 つのユーザを置いた。

提案手法と ProbCache で、遅延幅が小さい場合 (1ms-5ms) と遅延幅が大きい場合 (1ms-10ms) のそれぞれについて比較を行った。コンテンツのサイズは固定サイズで 1 とし、ルータのキャッシュサイズは隣接ルータの多いルータで 30 とし、それ以外を 10 とした。コンテンツ要求は 200 回の送信を行った。サーバが 1 つの場合と、隣接ルータが多いルータを新たなサーバとして追加しサーバを 2 つとした場合、隣接ルータが少ないルータを新たなサーバとして追加しサーバを 2 つとした場合の計 3 種類のシミュレーションを行った。

Sinet トポロジーにおけるシミュレーションの結果のうちサーバが 1 つの時のグラフを図 7 に、Geant トポロジーにおけるシミュレーション結果のうちサーバが 1 つの時のグラフを図 8 示す。また、Geant トポロジーについて、隣接ルータが多いルータをサーバとして追加した場合のシミュレーション結果を図 9 に、隣接ルータが少ないルータをサーバとして追加した場合の結果を図 10 に、それぞれ示す。

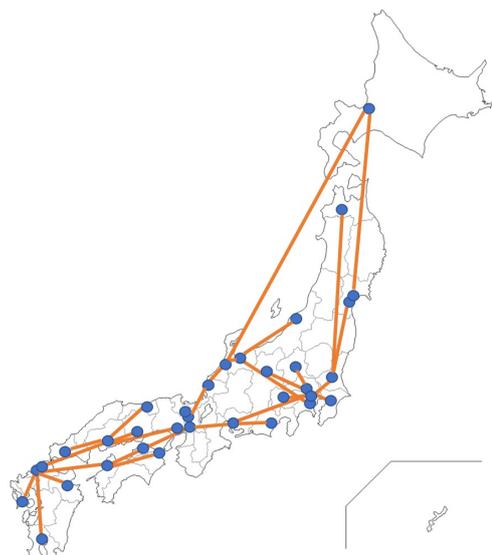


図 5 Sinet のトポロジー

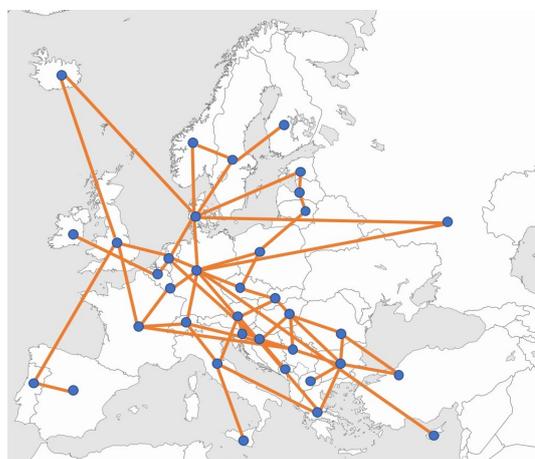


図 6 Geant のトポロジー

5.2 結果のまとめ

遅延時間幅が異なる提案手法と ProbCache でのキャッシュヒット率を ndnSIM のシミュレータを用いて結果を示した。結果のまとめを表 1 に示す。

表 1 シミュレーション結果まとめ

トポロジー	サーバ数	ProbCache との比較 [%]	
		遅延幅大	遅延幅小
Sinet	1	9.01	4.49
	2(隣接多)	16.0	12.4
	2(隣接少)	17.0	5.91
Geant	1	12.5	5.69
	2(隣接多)	28.8	11.7
	2(隣接少)	13.0	6.66

表 1 からすべての場合において提案手法のキャッシュヒット率は ProbCache でのキャッシュヒット率より高くなっていることがわかる。

またどの場合においても遅延時間幅が小さい時よりも遅延時間幅が大きい時のほうがキャッシュヒット率が高くなっている



図7 サーバ1つのときのシミュレーション結果 (Sinet)
上図:遅延幅大 (1-10ms), 下図:遅延幅小 (1ms-5ms)

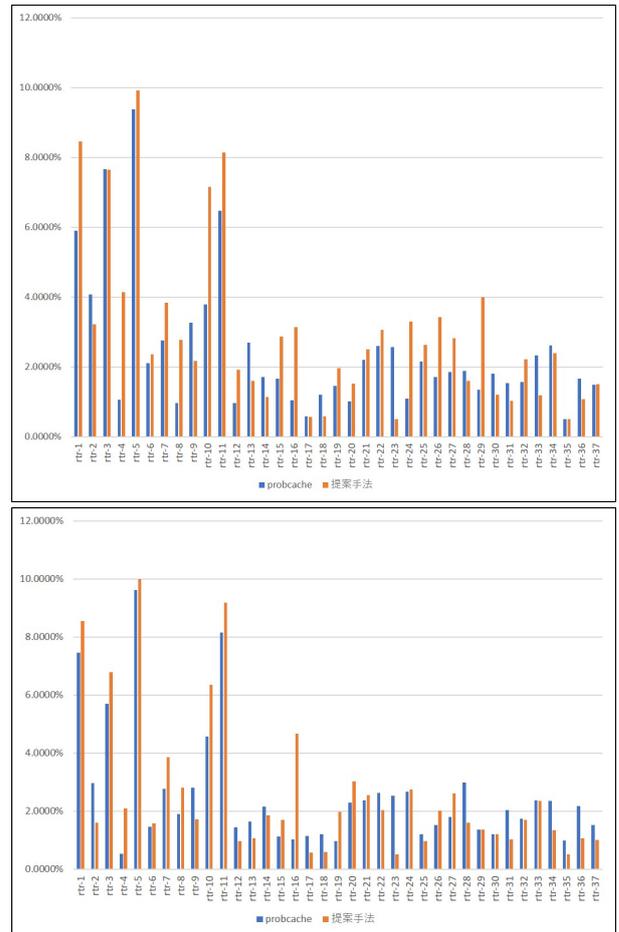


図8 サーバが1つのときのシミュレーション結果 (Geant)
上図:遅延幅大 (1-10ms), 下図:遅延幅小 (1ms-5ms)

ことがわかる。

5.3 考察

実際に使われているネットワークを用いてシミュレーションを行ったことから、その他のネットワークにおいても ProbCache よりもネットワーク遅延とキャッシュ容量を考慮した提案手法のほうが、キャッシュ機能が向上し、より効率の良いキャッシュができると考えられる。さらにはルータ間の遅延時間に差が大きくあるような広範囲に広がる大規模なネットワークではこの提案手法を用いることで高いキャッシュヒット率を得ることができる。

6. むすび

本稿では NDN のプロトコルにおいて、ネットワーク遅延とキャッシュ容量を考慮した、新しいキャッシュ方式の提案を行い、それを既存のアルゴリズムである ProbCache と比較し評価した。

本研究において遅延の違いや、サーバの数の違いによってのキャッシュヒット率の関係性は得ることができたが、サーバ同士の距離の違いによって具体的なキャッシュヒット率の変化の関係性までを得ることができなかったためその点が今後の課題として挙げられる。

文献

[1] Welcome to the geant website. [Online]. Available: <https://www.geant.org>, Last accessed on Feb. 4, 2020.

- [2] 学術情報ネットワーク sinet5. [Online]. Available: <https://www.sinet.ad.jp>, Last accessed on Feb. 4, 2020.
- [3] Hongseok Jeon, Byungjoon Lee, and Hoyoung Song. On-path caching in information-centric networking. In *2013 15th International Conference on Advanced Communications Technology (ICACT)*, pp. 264–267, Jan 2013.
- [4] Nikolaos Laoutaris, Hao Che, and Ioannis Stavrakakis. The lcd interconnection of lru caches and its analysis. *Performance Evaluation*, Vol. 63, No. 7, pp. 609 – 634, 2006.
- [5] Named Data Networking Project. Named data networking project website. [Online]. Available: <https://named-data.net>, Last accessed on Nov. 2, 2019.
- [6] Ioannis Psaras, Wei Koong Chai, and George Pavlou. Probabilistic In-network caching for information-centric networks. In *Proc. of ACM SIGCOMM Workshop on Information-Centric Networking (ICN-2012)*, pp. 55–60, August 2012.
- [7] 山本幹. コンテンツオリエンテッドネットワーク. 電子情報通信学会論文誌 B, Vol. J96-B, No. 6, pp. 589–604, 2013.
- [8] 高宮彰吾, 中里秀則. キャッシュ容量とネットワーク遅延に基づくキャッシュ方式. 2018 年電子情報通信学会総合大会予稿集, pp. B–8–17. 電子情報通信学会, 2018.
- [9] 山崎飛龍, 中里秀則. NDN におけるネットワーク遅延とキャッシュ容量に基づくキャッシュ方式. 2019 年電子情報通信学会総合大会講演論文集, pp. B–8–44, 3 月 2019.

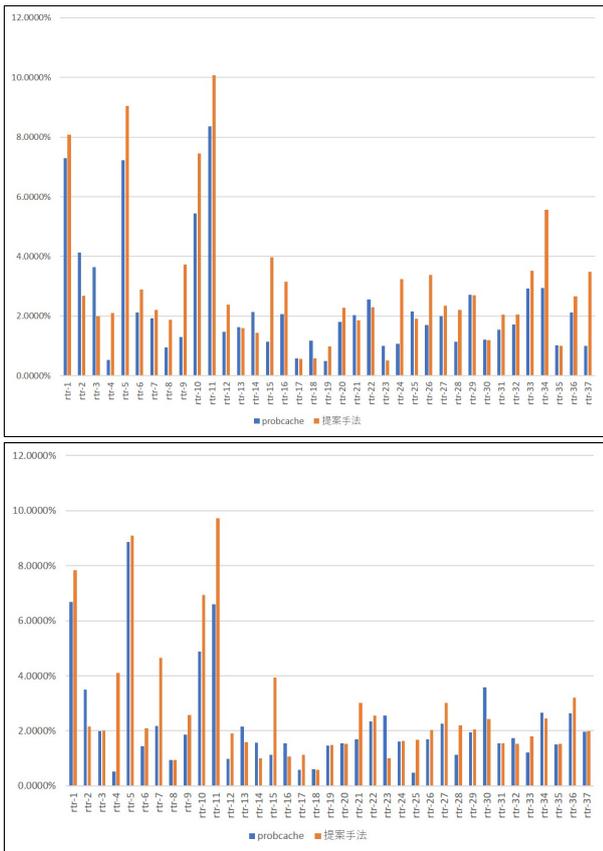


図 9 サーバ 2 つ (隣接ルータが多いルータをサーバとして追加)
(Geant)
上図:遅延幅大 (1-10ms), 下図:低遅延幅小 (1ms-5ms)

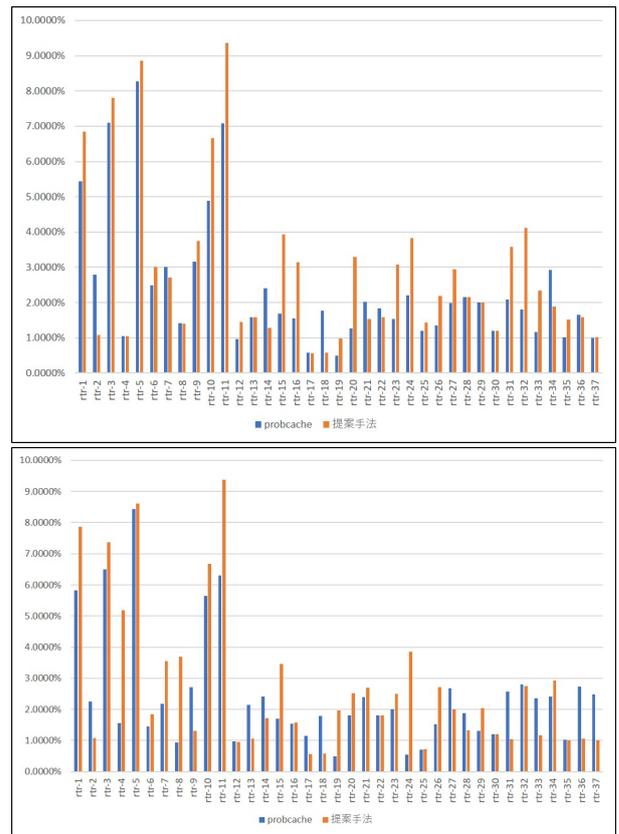


図 10 サーバ 2 つ (隣接ルータが少ないルータをサーバとして追加)
(Geant)
上図:遅延幅大 (1-10ms), 下図:遅延幅小 (1ms-5ms)