

2023年度

修 士 論 文

AS間トラフィックに着目したNDNの
動的ミラー配置法

指導教員: 上山 憲昭

立命館大学大学院 情報理工学研究科
情報理工学専攻 博士課程前期課程
計算機科学コース

学生証番号: 6611220047-0

氏名: CAO Xuheng

目次

概要	1
第 1 章 序論	2
1.1 背景	2
1.2 目的	2
第 2 章 関連研究	4
第 3 章 提案手法	6
3.1 他 AS のコンテンツのミラー化	6
3.1.1 他 AS のコンテンツの探知	6
3.1.2 コンテンツのミラー化	6
3.1.3 ミラーへの誘導	8
3.2 ミラーの最適配置	8
3.3 ミラーの動的生成	10
第 4 章 評価	12
4.1 評価条件	12
4.2 基礎の提案方式の有効性	13
4.2.1 コンテンツの人気が時間変動しない場合の評価	13
4.2.2 人気度の時間変化の実装	14
4.2.3 人気度の時間変化を考慮した性能評価	15
4.3 最適配置法の性能	19
4.4 動的なミラー生成法の性能	20
第 5 章 結論	25
謝辞	26
参考文献	28
学会発表リスト	28

概要

SNS やストリーミングの普及に伴い，インターネット上を流れるトラフィック量が急激に増えている．従来では良いネットワークサービスを提供するために，CDN (Content Delivery Network) などの緩和策を提案されていた．しかし根本となる TCP/IP はユーザにとって重要なコンテンツではなく，コンテンツの配信元であるホストに着目したパケット転送方式を採用している．そこで NDN (Named Data Networking) がコンテンツを効率的に配信する新しいネットワークアーキテクチャとして提案された．NDN にはネットワーク内のキャッシュからコンテンツを配信する機能を備えている．しかしコンテンツのキャッシュが近隣に存在しない場合，依然として遠方よりコンテンツを取得する問題が存在している．そこで本稿では AS 間トラフィック量の削減を目的に，動的にコンテンツのオリジナルのコピー（ミラー）を生成する方式を提案する．また，より現実的な状況を反映するため，現実のコンテンツの人気の変動を考慮した計算機シミュレーションにより，提案方式の性能を評価する．そして提案方式の有効性を確認する．

第1章 序論

1.1 背景

近年、SNS とストリーミング配信の急速な普及により、膨大な数のコンテンツが提供されている。また、単一のウェブページにアクセスするとき、実際には数百回～数千回のコンテンツへのアクセスが発生することがある [14]。そのため、ネットワーク上のトラフィックが急増している。それに伴いネットワークやパブリッシャへの負荷が急増し、QoS の低下が問題視されている。他に、複数の遠隔地における同一のコンテンツ配置による要求オーバーヘッドという問題も存在する [19]。

従来、コンテンツ配信の品質向上と配信コストの低減とのために Content Delivery Network (CDN) が広く用いられている。CDN は多数のキャッシュサーバで構成され、コンテンツをオリジンサーバの代わりに配信することで、コンテンツを提供するパブリッシャのオリジンサーバの負荷を軽減し、要求ユーザの近くからコンテンツを配信する。しかし現在のインターネットで用いられている TCP/IP は、ホスト間通信のために設計されており、全てのパケット通信はホストを特定し、宛先ホストに対してパケットが転送される。そのため、複数ホストに配置される同一コンテンツは異なる識別子になる [17]。なお、CDN はオーバーレイで構築されているものであるため、全ての性能を発揮することができない [11]。故に、CDN の構築及び利用には高いコストがかかる。

NDN (Named Data Networking)[18] は前述の問題を解消するために提案された。NDN においては、コンテンツを要求する際に用いるのはコンテンツを提供するホストの IP アドレスではなく、コンテンツの名前である。要求を転送する際、用いるフォワーディングテーブルは FIB (Forwarding Information Base) と呼ばれるものである。FIB は各コンテンツに対応するネクストホップという情報を記録しており、目標コンテンツの転送方向を示す役割を持っている。Interest パケットと呼ばれる要求パケットがノードに来るたび、要求コンテンツの名前に基づき、対応するネクストホップへ転送される。また NDN はネットワーク内のストレージを利用する機能を備えている。ノードが上記の Interest パケットを転送する前に、まず自身のキャッシュの中に、要求コンテンツをキャッシュしているかどうかをチェックする。もし、ノードが要求コンテンツをキャッシュしている場合、パブリッシャの代わりに Data パケットという応答用パケットでデータを返送する。また、ノードが Data パケットを転送する際に、キャッシュポリシーに基づき、Data パケットを自身のキャッシュに入れるかどうかを決定する。このような機能があるため、NDN はスケール性が高く、コンテンツ配信の性能も高いとされている。

しかしながら、NDN のキャッシュ機能にはまだ課題が残っている。NDN におけるネットワーク内のキャッシュを利用できるのは、Interest パケットの転送経路上に位置するノードに限定される。また、要求ユーザが存在する Autonomous System (AS) 内の Interest パケットの転送経路上に要求コンテンツがキャッシュされておらず、他の AS に存在するパブリッシャのところまで Interest が転送されることが多発する。

1.2 目的

前節では NDN のキャッシュがもたらすメリット及び残る課題を紹介した。そこで、キャッシュの利用効率を向上すれば、コストが高い AS 間トラフィックが削減でき、トラフィックの転送コストを大幅に下げることができる

[8]. 一番理想的な状態は、各 AS が自 AS 内のユーザの要求を自 AS 内部で完結させることである。そこで本研究では、配信要求を AS 内部で完結させるという視点から、従来のネットワークでコンテンツ配信性能を高めるために提案された CDN のコンセプトを参考にして、NDN において高コスト AS 間トラフィック量を削減し、NDN のキャッシュ性能を向上させる方式を提案する。

以下、2 章で NDN のキャッシュ性能を高めるための研究を紹介し、3 章で本研究の提案方式の詳細を述べる。そして 4 章で性能評価結果について述べ、5 章で全体をまとめる。

第2章 関連研究

前節で述べたように NDN はネットワーク上のキャッシュを活用することでコンテンツ配信を性能を向上するネットワークアーキテクチャである。そのため、キャッシュ性能は NDN の性能を大きく左右し、キャッシュ性能に関する研究が数多くみられる。

[4] では、TCP/IP で重要な役割を果たしている Domain Name System (DNS) のコンセプトを参考にして、常に最適なキャッシュにアクセスできる手法が提案されている。著者らは NDN におけるキャッシュ発見は Interest パケットの転送経路上しか行えず、キャッシュの利用率が低いことを指摘し、近くのノードにキャッシュされているキャッシュを無視する可能性が存在する問題点に着目している。もし全ての Interest パケットが最寄りのキャッシュへ転送されるようになったら、キャッシュ性能が大幅に向上できるという考えから、コンテンツ名とキャッシュ位置と紐づけた NDN における名前解決手法を提案している。具体的には、各 Internet Service Provider (ISP) 内に一つ集中型の追跡サーバを設置し、常時 ISP 全域のキャッシュ情報を収集する。そして、要求が発生するたびに Interest パケットの情報を基づき、追跡サーバに問い合わせし、要求コンテンツのキャッシュの存在を確認する。もし ISP 内部に要求コンテンツのキャッシュが存在すれば、追跡サーバは一番近いキャッシュを保持するノードまでの転送経路を計算し、返答として問い合わせしたノードへ返送する。その場合、ノードは通常のルーティングをせずに、Interest パケットを要求コンテンツのキャッシュを所持している最寄りのノードへ、最短経路を沿って転送する。

しかし、追跡サーバの設置コストが大きい上に、ネットワーク内のキャッシュ情報を収集するオーバーヘッドも高い。また、スケール性を高めるためにはルーティングが自律分散である必要があり、集中型サーバを用いると、NDN のスケール性が損なわれる。

[7] では NDN で CDN を構築するために階層化したキャッシュ制御方式が提案されている。この研究ではネットワーク上のノードを3種類に分け、それぞれ異なる役割を持たせることでキャッシュ利用率を向上させる。レイヤ1のノードはCDNのキャッシュサーバと同じ役割を果たし、CDNのような機能が動作する保証となる。言い換えれば、レイヤ1のノードが必ずコンテンツのキャッシュを保持する。レイヤ2のノードは人気コンテンツをキャッシュしているノードである。このレイヤのノードは人気コンテンツの存在を周辺ノードへ示し、要求を誘導することで、レイヤ1のノードより近いところから人気コンテンツの配信を実現する。レイヤ3のノードは通常の NDN ノードである。すべての要求は下層レイヤから上層レイヤへ向かって転送されるようにすることで、近くのキャッシュをより早く発見することを実現した。

しかし、階層化したキャッシュ制御でキャッシュの利用率と性能が向上するものの、CDN を構築するための構造であるために NDN 本来のスケール性が損なわれる。一方、NDN を用いれば、一部のトポロジにおけるキャッシュ性能は CDN より高いとされており [13]、NDN のキャッシュ性能を活用することで CDN の必要性も無くなると考えられる。

[12] では、AS 間トラフィックに着目した自律分散型の方式が提案されている。この研究では AS 内部のノード間に AS 外部のコンテンツに関するアクセス情報を交換する。ノードは各々その情報に基づき、キャッシュの位置を調整し、キャッシュ配置の最適化を行う。

ノードの協調によりキャッシュの最適化を実現されたものの、キャッシュの移動や置換が頻繁に発生するため、

ストレージコストが高いという問題が存在する．またこの方式はツリー型のトポロジを前提にした方式であるため，汎用性が欠けている．

上述の問題点を踏まえ，本研究では NDN のキャッシュ性能及びキャッシュ発見能力を向上することでコストの高い AS 間トラフィックを削減するためのミラーの動的配置法を提案する．ミラーとはノードに長時間保持され，配信されるキャッシュのことである．人気コンテンツをミラーにし，トラフィックをミラーへ誘導することでキャッシュ発見の効率を向上する．なお，ミラーは長期間にわたって存在するものであるため，置換頻度が低く，ストレージコストが抑制できる．また，提案手法はミラー化するコンテンツを選定するための情報はノード間の交換が要らない．そのため，協調によるオーバーヘッドも抑制できる．

第3章 提案手法

本章では一部の外部コンテンツをミラー化し、自 AS 内のトラフィックをミラーへ誘導することで、AS 間トラフィック量を減らす方式について述べる。まず 3.1 節で提案方式の根幹部分となる他 AS のコンテンツをミラー化する方式を述べる。次に 3.2 節と 3.3 節で、提案方式の性能を向上するための拡張方式について説明する。

3.1 他 AS のコンテンツのミラー化

本節ではパブリッシャが AS の外部に存在するコンテンツの中で、AS の内部に需要のあるコンテンツをミラー化する方式について述べる。ミラーとは新しいコンテンツのコピーを空き容量のないキャッシュに入れようとし、置換対象を選ぶとき、選択対象から外れたキャッシュのことをいう。NDN のノードは実装している置換ポリシーに基づき、キャッシュを入れ替えている。しかし、頻繁な置換はストレージコストが高い上、人気のあるコンテンツを置換してしまう可能性が存在する。ミラーを用いれば、この状況を防げる。また、コンテンツのコピーをミラーにすることを以下ミラー化と言う。

3.1.1 他 AS のコンテンツの探知

1 章で述べたように、本研究では AS 間トラフィックに着目したキャッシュ制御を提案する。具体的には元々 AS 外部へ転送される要求を AS 内部で解決すれば、AS 外部へ転送しなくなり、AS トラフィックの削減を実現するという考えである。なお、外部への要求を全て AS 内部で解決することが現実ではないため、人気コンテンツへの要求のみ解決することが望ましい。

それを実現するため、外部へ転送される要求の目標コンテンツ及びコンテンツの人気度を探知する手法が必要となる。なお、要求を外部へ転送されることは目標コンテンツのパブリッシャが AS の外部に存在することを意味するため、以下このようなコンテンツを外部コンテンツと言う。AS の外部へ転送される要求が AS を出るとき、必ず AS の境界に存在するゲートウェイルータを経由する。そこで、ゲートウェイルータに自身を経由して外部へ転送される要求の目標コンテンツ名をキャプチャし、対応するアクセス数を記録させれば、外部コンテンツに対する需要が把握できる。なお、各コンテンツのアクセス数から、AS 内部のユーザにとって人気の高い外部コンテンツも分る。また、本研究では AS のゲートウェイはある程度のスペックを有するマシンと想定するため、要求数の集計のための性能は十分有すると考える。

3.1.2 コンテンツのミラー化

外部コンテンツのアクセス情報を把握したら、ミラー化する対象を選定する。AS 内部のキャッシュ容量は無限ではないため、ゲートウェイは AS 内部のキャッシュ容量に合わせて、人気度上位 $K\%$ の外部コンテンツのみ選択する。また、過去の人気コンテンツをミラー化の対象として選択することを防ぐために、過去一定期間以内のアクセス情報のみ参照する。

ゲートウェイがミラー化するコンテンツセットを決定したら、図 3.1 のステップ 1 が示すよう、自 AS 内のノートヘミラー化の要請を送信する。しかし、NDN はプルベースの通信モデルを採用しており、プッシュ操作が対応されていないため、特定のノードへ要請パケットを送るために、ブロードキャストが必要となる。ブロードキャストを採用すると、ミラー化の要請を送るだけで莫大なトラフィックが発生する問題が生ずる。この問題を解決するために、NDN 上で実現した pub/sub モデル [10, 5, 16, 15] を導入する。AS 内部のノードにゲートウェイのミラー要請を購読させることで、ゲートウェイが AS 内部のノードヘミラー化の要請をプッシュすることを可能にする pub/sub モデルを実現するためのオーバーヘッドが高いと言われている。しかし本提案手法では、購読者が AS 内部のノードに限定され、購読対象もゲートウェイしかないため、導入コストが低いと考えている。この要請パケットにはミラーを立てる時に必要な情報が記載されている。このパケットを受け取ったノードはパケット内の情報に基づき、協調しミラー化する。そのときノード x のキャッシュにまだ空き容量があれば、図 3.1 のステップ 2-a が示すよう、ノード x は予め該当コンテンツを要求し、キャッシュし、ミラー化する。ノード x のキャッシュの空き容量が不足するときは、図 3.1 のステップ 2-b が示すよう、該当コンテンツをキャッシュせずに周辺ノードへ要請を転送し、要請を受けたノードは同じ処理を行う。それで、自 AS 内の全てのノードが所持するストレージが利用できる。

上述の一連の処理を一定周期 T おきに行う。なお、ミラー化したコンテンツは常々ミラー化された時と同じほどの人気度を維持することが不可能であるため、古いミラーを定期的に削除する必要がある。そこで、ミラーのライフタイムもミラー化周期と同じ長さとする。それで、新しいミラーを立てるたびに、前の周期で作ったミラーは期限を超え、一般のキャッシュとして置換されることが可能となる。

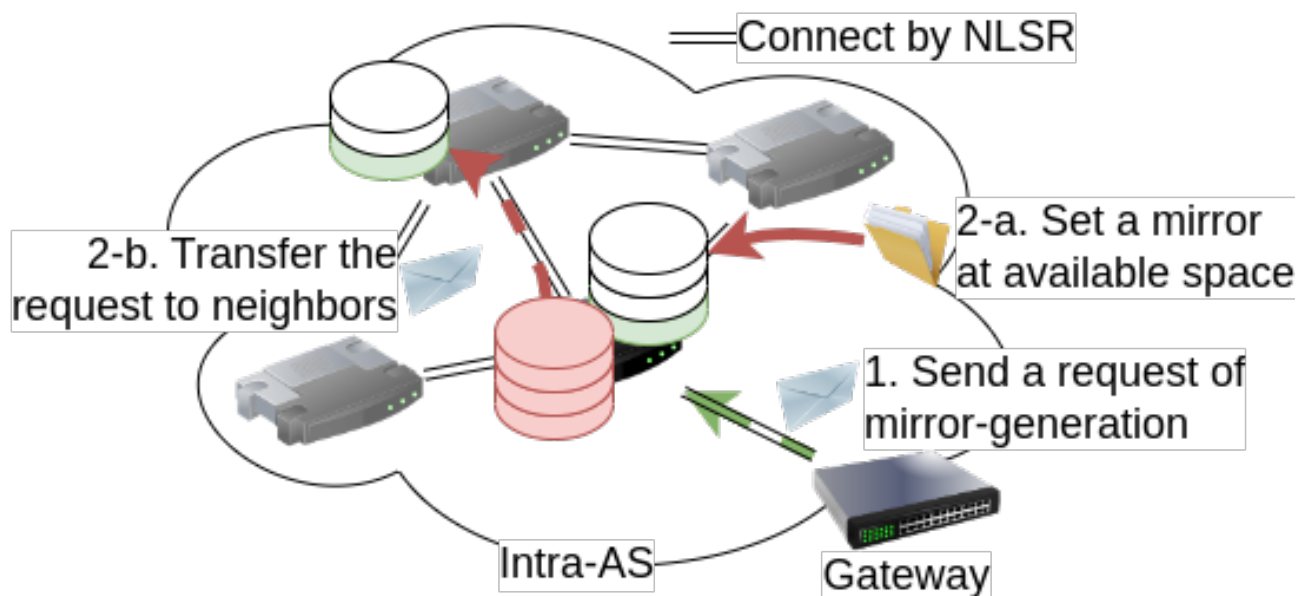


図 3.1: Sending mirror request from gateway to selected routers in AS

3.1.3 ミラーへの誘導

ミラー化のみを行うと、依然として転送経路上のキャッシュしか利用できず、転送経路外のミラーを利用することが不可能である。そのためには自 AS 内の要求が新たに作成したミラーへ転送されるようにする必要がある。本節では、それを実現するための手法を紹介する。

NDN において、要求パケットの転送方向はフォワーディングテーブルである FIB に基づいて決める。FIB は要求パケットの目標コンテンツに対応するネクストホップを探し、転送する。そこで、FIB にミラーへの転送方向を教え、ミラーに対応するネクストホップを更新することで、転送方向を変更する。具体的にはミラー化を行ったノードが AS 全域へミラーの存在を広告する方式である。すなわち、AS 内の全ノードに対し、ミラー化コンテンツの名称を保存するパケットを NLSR[9] を用いてブロードキャストする。本ブロードキャストのパケットを受け取ったノードは、パケット内に記載されたコンテンツ名に対応する FIB 上のエントリのネクストホップを、本パケットが到着したポート番号に変更することで、転送パスを更新する。また、ミラーは有効期限が存在する。ミラーが消失してもパケットのフォワーディングが正常に行えるよう、FIB 内の本ミラーのエントリにも同一の有効期限を付与し、元のネクストホップを保存する。その結果、図 3.2 に示すように、ミラー化したコンテンツに対する要求は AS 外部へ転送されずに AS 内に生成したミラーへ転送される。これで、本コンテンツに対する AS 内ユーザからの要求は全て AS 内部で完結する。またコンテンツをミラー化することにより、ユーザの近いところからコンテンツを配信することとなり、要求遅延と AS 内トラフィック量の低減も期待できる。

3.2 ミラーの最適配置

3.1 節では自 AS において、人気のある他 AS のコンテンツを AS 内部のノードでミラー化し、AS 内部の要求を内部で完結させる方式を提案した。しかし、提案手法により、元々ゲートウェイへ向かう要求パケットの転送方向が目標コンテンツのミラーを所持しているノードに変更された。そこで、ミラーの設置する場所によっては AS 内部のトラフィック量を増やす可能性が存在する。そのような状況を防ぐため、本節ではミラー配置場所の最適化アルゴリズムを提案する。

ミラー配置の最適化はミラー化することによる自 AS 内のトラフィック増加を防ぐためである。そのため、一番理想的な状態は自 AS 内の要求パケットがミラーまでの平均距離を最小化できることである。また、本研究では一つのコンテンツに対して複数のミラーを設置することにより、ユーザの要求遅延を更に削減することも考えている。故に最適配置法の目標は最寄りのミラーまでの平均距離を最小化することである。

そこで、最適配置アルゴリズムに求める出力は、入力されたトポロジから、1 個以上のノードの有するノードセットの中で、最寄りのノードへの平均距離が一番小さいノードセットである。もしノードセット内のノード数が 1 個の場合、求める結果は近接中心性の一番高いノードを探し出すことになる。そのため、本稿では以下「ノードセット内のノードの中で最寄りのノードへの平均ホップ長の逆数」ということをこのノードセットの近接中心性と定義する。以下はノードセットの近接中心性の計算方法である。

- N : トポロジ内のノード数
- $i \in I$: 目標ノードセット I のノード
- $j \in J$: ノードセット I に含まれないノードセット
- nI : ノードセット I に含まれるノードの数
- d_{ij} : I の中で j に一番近いノード i から j までの距離

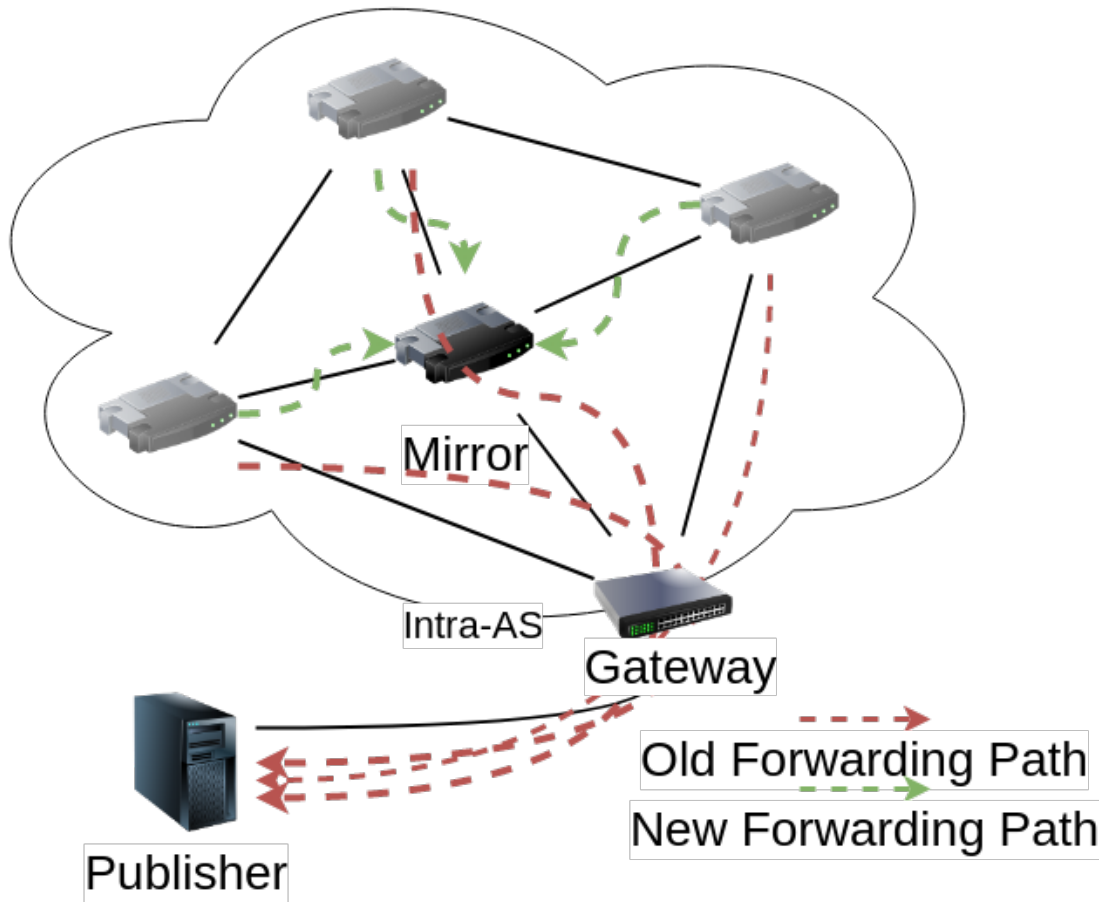


図 3.2: Changes Interest-forwarding routes to mirror in intra AS from publisher in external AS

- ノードセットの近接中心性 (CCNS: Closeness Centrality of Node Set):

$$CCNS = \frac{1}{\frac{1}{N-nI} \sum_{j=1}^N d_{ij}}$$

従来では、近接中心性の一番高いノードを求めるためにトポロジ内の全てのノードの近接中心性を求めて比較する必要がある。近接中心性の一番高いノードセットを求めることも同じである。しかし、全ての近接中心性を求める計算量は膨大なため、オーバーヘッドが大きい。そこで本稿では、郵便ポスト最適配置問題を参考し、低い計算量で近似的な最適解が求められるよう、Algorithm 1 に示す最適配置アルゴリズムを提案する。

Algorithm 1 Optimally selecting nodes placing mirrors

-
- 1: Input: n , mirror count and G , topology
 - 2: Randomly selects n candidate nodes C in G
 - 3: Derives H , average hop length to closest node in C from each node in G
 - 4: Finds node C_n adjacent to any node x of C with condition that we can reduce H if we replace C_n with x
 - 5: Replaces C_n with x and return step 3 if C_n exists
 - 6: Completes algorithm after outputting C as results if C_n does not exist
-

最適配置先を上述のアルゴリズムで求めたら、3.1.2 節で述べたよう、ゲートウェイから pub/sub モデルを用いてミラー化の情報を最適配置先に送信する。このパケットを受け取ったノードが行う動作は以下のようになる。

1. 自身のストレージにミラー用の空き容量があるかどうかをチェックする。
2. 空き容量があるとき、該当コンテンツをミラー化し、期限を付けてからキャッシュに入れる。
3. 空き容量がないとき、キャッシュ内部で該当コンテンツより人気度（要請パケットに記載されている）が低いミラーを探す。
4. 人気度の該当コンテンツより低いミラーが存在する場合、一番古いミラーを置換する。
5. 存在しない場合、要請パケットを隣接ノードへ転送し、要請を受けたノードが同じ処理を行う。

これで、高い人気度のあるコンテンツは常に最適配置先の周辺に配置されるようになる。

3.3 ミラーの動的生成

3.1.1 節では固定な周期で自 AS において人気のある他 AS のコンテンツを探知する手法を提案した。この方式を用いれば、低いオーバーヘッドで定期的に新しい人気コンテンツを探知し、古いミラーを置換することができる。しかし、コンテンツの人気度は常に変化しており、各々の変化速度も異なっている。そのため、固定的な周期でコンテンツの人気度を検知するだけでは、リアルタイムの人気度変化を反映できず、人気度を失いかけているコンテンツをミラー化の対象として選ぶ可能性が存在する。そのようなコンテンツのミラーは性能が低下する原因となる。本節では、常時に他 AS のコンテンツの人気度を検知し、あるコンテンツが急上昇し始まったら、そのコンテンツをミラー化する動的ミラー検知法を提案する。

人気のあるコンテンツへのアクセス数は一般コンテンツよりはるかに多いことから、人気のあるコンテンツのアクセス数は他のコンテンツのアクセス数から大きく外れた外れ値と言える。言い換えれば、全てのコンテンツのアクセス数から外れ値を検知することで、人気コンテンツを探知することが可能となる。そこで利用するのは Z スコア [3] である。Z スコア法を用いれば、ある値がどれほどデータセットから離れているかを評価できる。ゲートウェイで収集した全ての他 AS のコンテンツに対する要求パケットの数をデータセットとし、各コンテンツのアクセス数に対応する Z スコアを常時に計算する。そして、あるコンテンツの Z スコアが予め設定した閾値を超えたら、そのコンテンツをミラー化の対象として選んで、3.1.2 節で述べた方式でミラー化を行う。

また、動的にミラーを生成する場合はミラーの置換を行う必要がある。そこで、ゲートウェイに全てのミラーの生成時間及び位置を把握させる。そして、ゲートウェイが一番古いミラーを所持するノードにミラー化の要請

を送る．ミラー容量の足りないノードがミラー化の要請パケットを受け取った時，新しいミラーで古いミラーを置換する．これで，常に人気上昇し始めたコンテンツをミラー化すること，及び古くなったミラーを動的に削除することが可能となる．

第4章 評価

本稿では NDN において、コストの高い AS 間トラフィック量を削減する動的ミラー構築方式を提案した。本提案方式の有効性を確認するために、本節では計算機シミュレーションによりキャッシュヒット率や AS 外ホップ長を評価する。本研究では AS 間トラフィック削減に着目し、AS 外のコンテンツのみをミラー化対象としている。そのため、シミュレーションの環境では外部コンテンツしか存在しないとする。また、ネットワーク上のコンテンツの人気度分布は Zipf 分布に従うことが知られているため [6]、本稿でも Zipf 分布に基づき各コンテンツの要求比率を設定する。

4.1 評価条件

以下は性能評価で用いる用語の定義である。

- ミラーサイズ比: 各ノードのキャッシュ容量のうちミラーに割り当てる容量の割合
- ミラー数: 一個の外部コンテンツに対して構成するミラーの数
- 平均 AS 間経路長: 全ての Interest パケットが要求ユーザ AS の外部で転送された平均ホップ数
- キャッシュヒット率: ユーザの配信要求に対して、パブリッシャのオリジンサーバ以外の場所、すなわちルータのキャッシュか、ミラーからデータを配信できた割合

また、キャッシュ制御方式として以下のポリシーを各々、用いた場合の評価を行う。

- キャッシュポリシー:
 1. LCE: Data パケットが経由する全てのノードでコンテンツをキャッシュ
 2. PROB: Data パケットの経路上の各ノードで、パブリッシャからのホップ距離に反比例する確率でコンテンツをキャッシュ
 3. UNIPROB: 固定確率で、Data パケットの経路上の各ノードで各々、コンテンツをキャッシュ
- 置換ポリシー:
 1. LRU: 最後に要求されてからの経過時間が最大のコンテンツを削除
 2. FIFO: 最も古い時刻にキャッシュされたコンテンツを削除

計算機シミュレーションで用いたデフォルトの各種設定値を表 4.1 に示す。

表 4.1: Simulation Configurations

Parameter	Value
Topology	At Home Network[1]
Zipf parameter	0.8
Simulation Time	200,000 requests
Number of gateways	3
Ratio of mirrored contents, K	0.1
Number of contents	10,000
Cache size at each node	100
Mirror Building Cycle	per 50,000 requests

4.2 基礎の提案方式の有効性

本節では最適配置法及び Z スコア法が実装されていない提案手法の性能を評価する。

4.2.1 コンテンツの人気が時間変動しない場合の評価

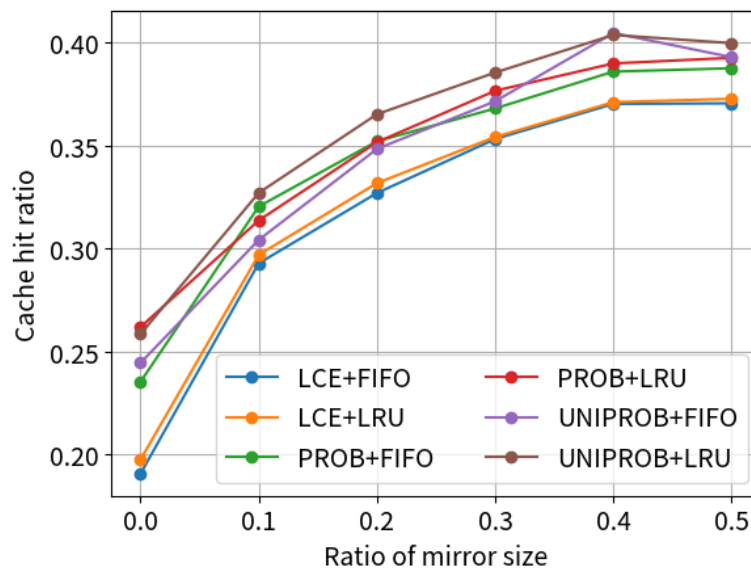


図 4.1: Cache hit rate against mirror size ratio

まずはコンテンツの人気が時間的に変動しない場合の、提案方式のキャッシュヒット率と平均 AS 間ホップ長を評価する。図 4.1 に、様々な各キャッシュポリシーにおける、提案方式を用いた場合のキャッシュヒット率を、着目 AS 内のミラーサイズ比に対して示す。ミラーサイズ比が 0.0 の場合は、提案方式を用いない場合に相当する。このグラフから、ミラーサイズ比の増加に伴い、キャッシュヒット率が上昇するが、ミラーサイズ比が 0.4 を超えたあたりから、キャッシュヒット率の上昇が見られなくなったことがわかる。これは、キャッシュとして用いるこ

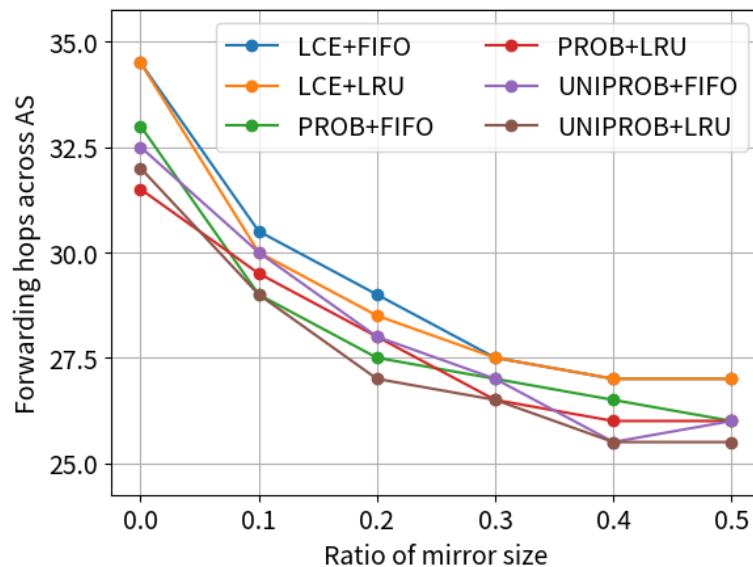


図 4.2: Average inter-AS hop length against mirror size ratio

とができるメモリのサイズが小さくなることが要因と考える．図 4.2 に，ミラーサイズ比に対して AS 間平均ホップ長を同様にプロットするが，提案方式を用いることで，AS 間トラフィックの削減効果はミラーサイズ比の増加に伴い，やはり増加することが確認できる．以上のことから，提案方式の有効性が確認できる．

4.2.2 人気度の時間変化の実装

4.2.1 節では，各コンテンツの人気度が固定で時間的に変化しない場合において，提案方式の有効性を確認した．しかしコンテンツの人気度が不変であるという前提は非現実的であるため，より現実に近い評価ができるよう，コンテンツの人気度変化を考慮した場合について評価する．本節ではシミュレーターで実装された人気度の時間変化の仕組みについて説明する．

まず，Google API から取得した著名映画の検索回数の時系列データを正規化し，これら映画コンテンツの人気度変化を示す時系列データとみなす．そして複数人気度時系列データを，以下の 3 種類の変化パターンにまとめる．

1. コンテンツがリリースされた時は高い人気度であるが，すぐに人気度が低下する．
2. コンテンツがリリースされた時は高い人気度で，かつ長い期間の間，高い人気を保つ．すなわち人気は緩やかに低下する．
3. コンテンツがリリースされたあとは人気の変化があるものの，リリースされた時の人気を維持する．

新しいコンテンツを生成するたびに，前述の 3 種類の人気度変化パターンからランダムに一つ選択する．同時にコンテンツのライフタイム，つまり需要がなくなるまでの時間を一定範囲内の乱数を用いて設定する．常時に存在するコンテンツ数を 10,000 個ぐらいに維持するため，コンテンツのライフタイムをおおよそ 10,000 個後のコンテンツが生成される時に自身のライフタイムが切れ，削除されるように設定する．その後，選択した人気度変化パターンに基づき，各コンテンツに対し人気度変化の時系列データを作成する．また，すべてのコンテンツの人気度変化が同じ傾向とならないよう，乱数で各コンテンツの時系列データを微調整する．

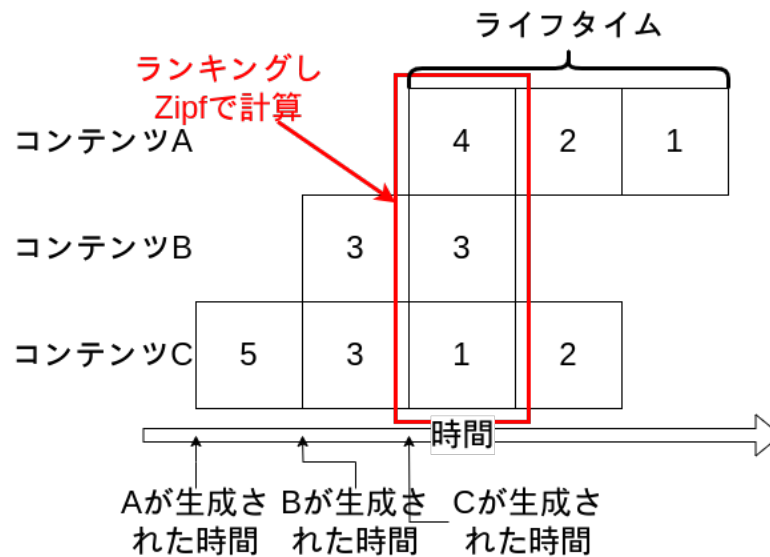


図 4.3: Content generation with popularity change

各コンテンツの時系列データを生成したら、Zipf パラメタで各々の人気度を計算する。そして、各コンテンツの人気度変化データに基づき、人気ランキングをタイムスロットごとに決める。具体的には図 4.3 が示すよう、コンテンツが生成された時間に合わせて並ぶ。次は同じタイムスロットにある各コンテンツの人気度データに基づき、ランキングする。最後はランキングに基づき、各コンテンツの各タイムスロットの要求確率を Zipf 分布で設定する。また、シミュレーションの進行にともない、新しいコンテンツを生成し、古いコンテンツを削除することで、現実に近い人気度変化をシミュレータに実装した。

4.2.3 人気度の時間変化を考慮した性能評価

本節では人気度の時間変化を実装したシミュレータで提案方式のキャッシュヒット率と平均 AS 間ホップ長を評価する。まず人気度の時間変化に関する値の定義を説明する。

- コンテンツ生成頻度：1 秒間で生成するコンテンツの数である。この値が高いほど高い頻度で新しいコンテンツを生成し、人気度の変化も激しくなる。

図 4.4 にミラーサイズ比を 0.4 としたときの、コンテンツ生成頻度に対し、提案方式のキャッシュヒット率、及び従来方式のキャッシュヒット率を示す。グラフの中で、実線が示すのは提案方式の結果である。それに対して、点線が示す結果は従来方式の結果である。

グラフが示すよう、コンテンツ生成頻度の増加に伴い、ヒット率は低減した。原因として考えられるのは、ミラーの構築速度がコンテンツの人気度変化に追いつかないことである。そのとき、ミラーの鮮度が消えやすくなるため、キャッシュ性能も下がる。しかしながら、コンテンツの人気度変化が激しい場合も、提案方式を用いることで従来方式より優れる結果が得られた。これで、人気度の変化を考慮する場合でも、提案方式は有効であることを確認した。

本節以後の評価はすべてコンテンツ生成頻度を 2 に設定した時の結果となる。また、4.2.1 節と同じく、横軸のミラーサイズ比が 0.0 の時、及びミラー数が 0 の時が提案方式を用いない場合に相当する。

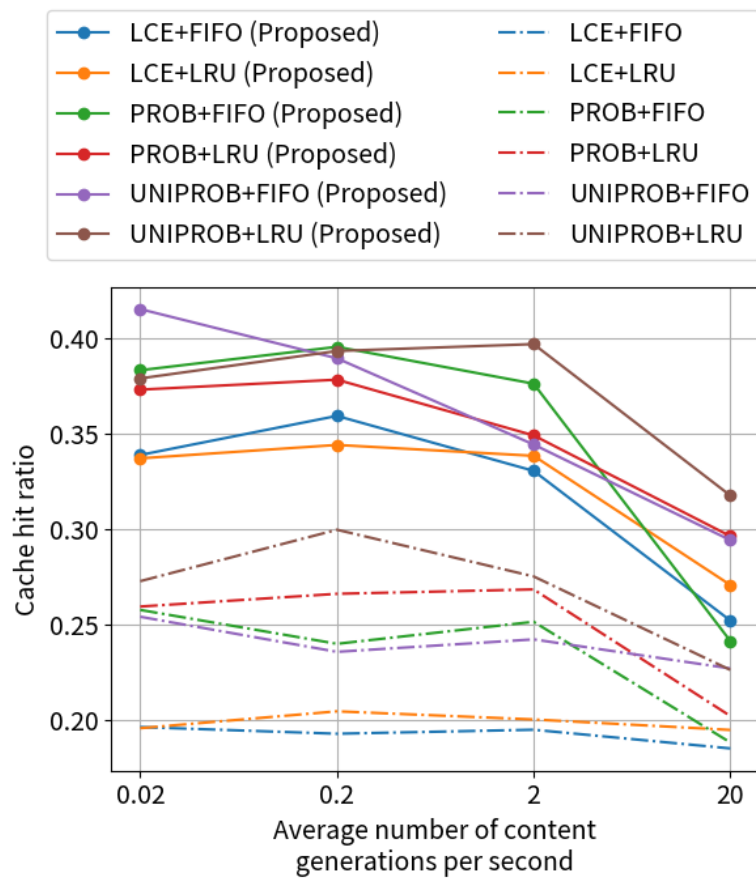


図 4.4: Cache hit rate against relative cycle length of mirror generation

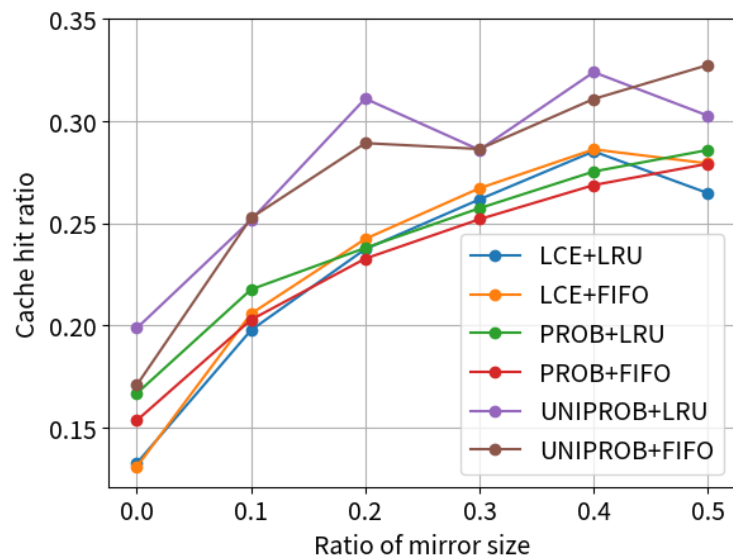


図 4.5: Cache hit rate against mirror size ratio

図4.5にキャッシュサイズ比に対してキャッシュヒット率を示す。図4.1に示したように、やはりキャッシュサイズ比の増加に伴いヒット率は増加するが、ミラーサイズ比が0.4を超えたあたりから、キャッシュヒット率の増加が見られなくなる。

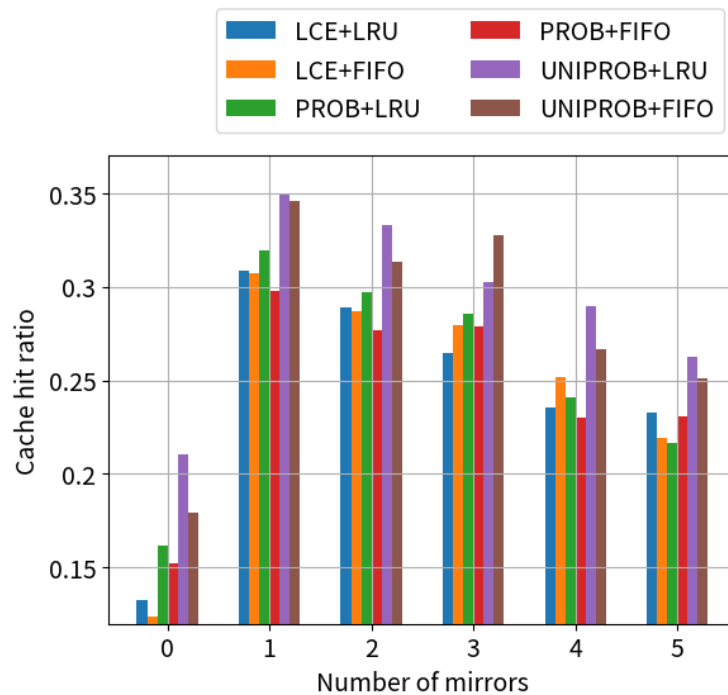


図 4.6: Cache hit rate against number of mirrors with 50% mirror size

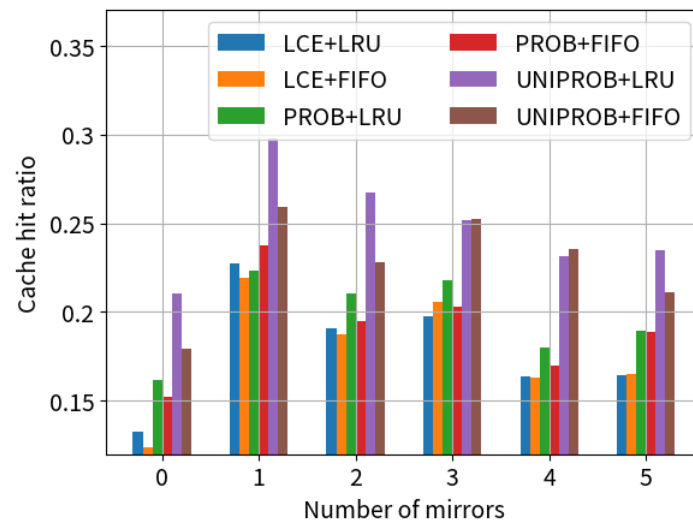


図 4.7: Cache hit rate against number of mirrors with 10% mirror size

図4.6はミラーサイズを40%に設定したとき、ミラーサイズミラー数に対するキャッシュヒット率を示す。ただ

し AS 内の全ルータでミラーを収容できるルータのキャッシュ容量は一定なので、ミラー数の増加に伴い、ミラー化の対象とするコンテンツ数は減少する。提案方式を用いない場合 (ミラー数が 0) の場合と比較して、提案方式を用いることでヒット率が大幅に増加すること、またミラー数の増加に伴いヒット率は低下することが確認できる。これはミラー数の増加に伴い、ミラー化できるコンテンツ数が減少し、ミラーの多様性が低減したためである。しかし、ミラー数を増やせば、ことなる場所にミラーを設置することとなり、ユーザからミラーまでの平均距離が短くなる。その場合、ミラーを取得する遅延が減少する。なお、ミラーの数を増やすことで、ミラーへの負荷を軽減することもある。

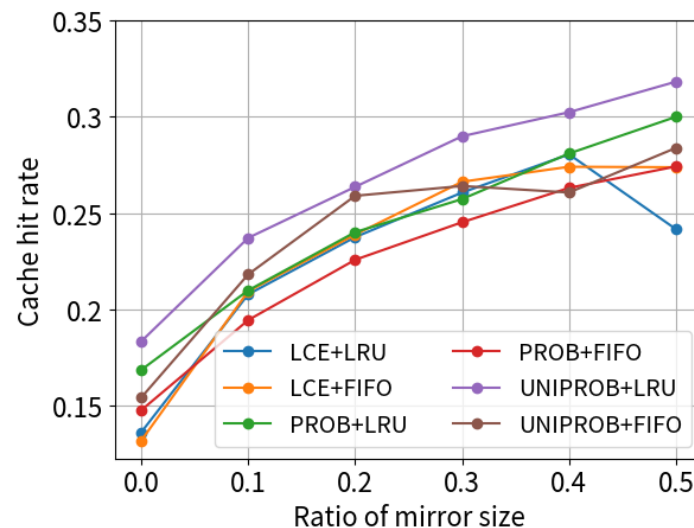


図 4.8: Cache hit rate against mirror size ratio in Verio

また、図 4.7 はミラーサイズを 10% に設定したとき、ミラーサイズミラー数に対するキャッシュヒット率を示す。この図から、ミラーサイズを減らしても、提案手法は効果があるということがわかる。

ここまでの評価はすべて Ladder 型のトポロジ、すなわち次数の高いノードが存在しないトポロジである At Home Network における評価である。しかし、ネットワーク上のトポロジは色々な種類があり、異なるトポロジでの効果も確認する必要がある。そこで、高次数のハブノードが存在する Hub and spoke (H&S) 型のトポロジである Verio[2] のトポロジを用いて、提案方式の性能を評価する。

図 4.8 はミラーサイズ比に対するキャッシュヒット率を示すグラフである。図 4.9 はミラー数に対してキャッシュヒット率を示した。グラフから、H&S 型のトポロジでも、提案手法を用いるときの性能向上が確認できた。これで、異なるトポロジにおいても、提案手法は有効であることを確認した。

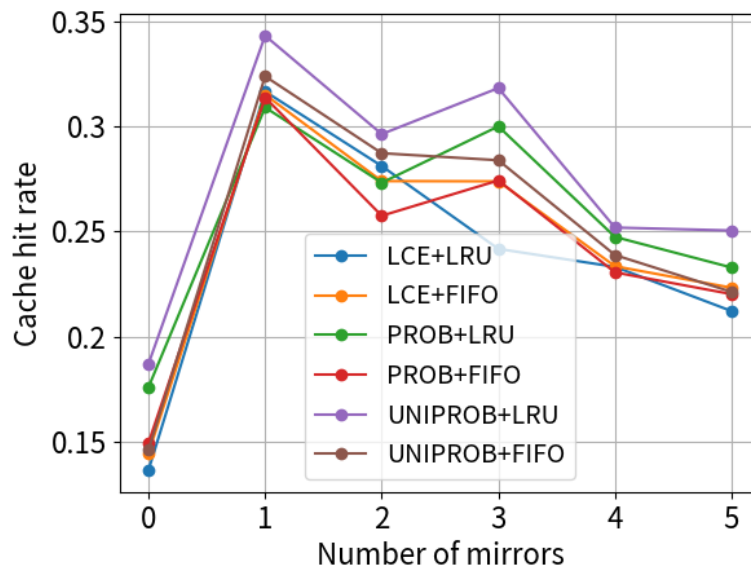


図 4.9: Cache hit rate against mirror count in Verio

4.3 最適配置法の性能

3.2 節ではミラーの最適配置先を選出するアルゴリズムを提案した。本節ではそのアルゴリズムの有効性を評価する。3.2 節で述べたよう、最適配置アルゴリズムは最寄りのミラーまでの平均距離を最小化するノードセットを選出するアルゴリズムである。そのため、アルゴリズムの結果はいかに最小化できるかを評価することで、アルゴリズムの有効性が確認できる。そこで、提案したアルゴリズムを異なるトポロジを用いて千回実行し、選出したノードセットの近接中心性の平均値を算出する。そして、最適解の近接中心性を算出し、アルゴリズムの結果と比較することで、提案手法の有効性を確認する。

図 4.10 は異なるトポロジにおける実行結果を示した。横軸はノードセット内のノード数、つまりミラーの数を意味する。縦軸はノードセットの近接中心性 (CCNS) を意味する。

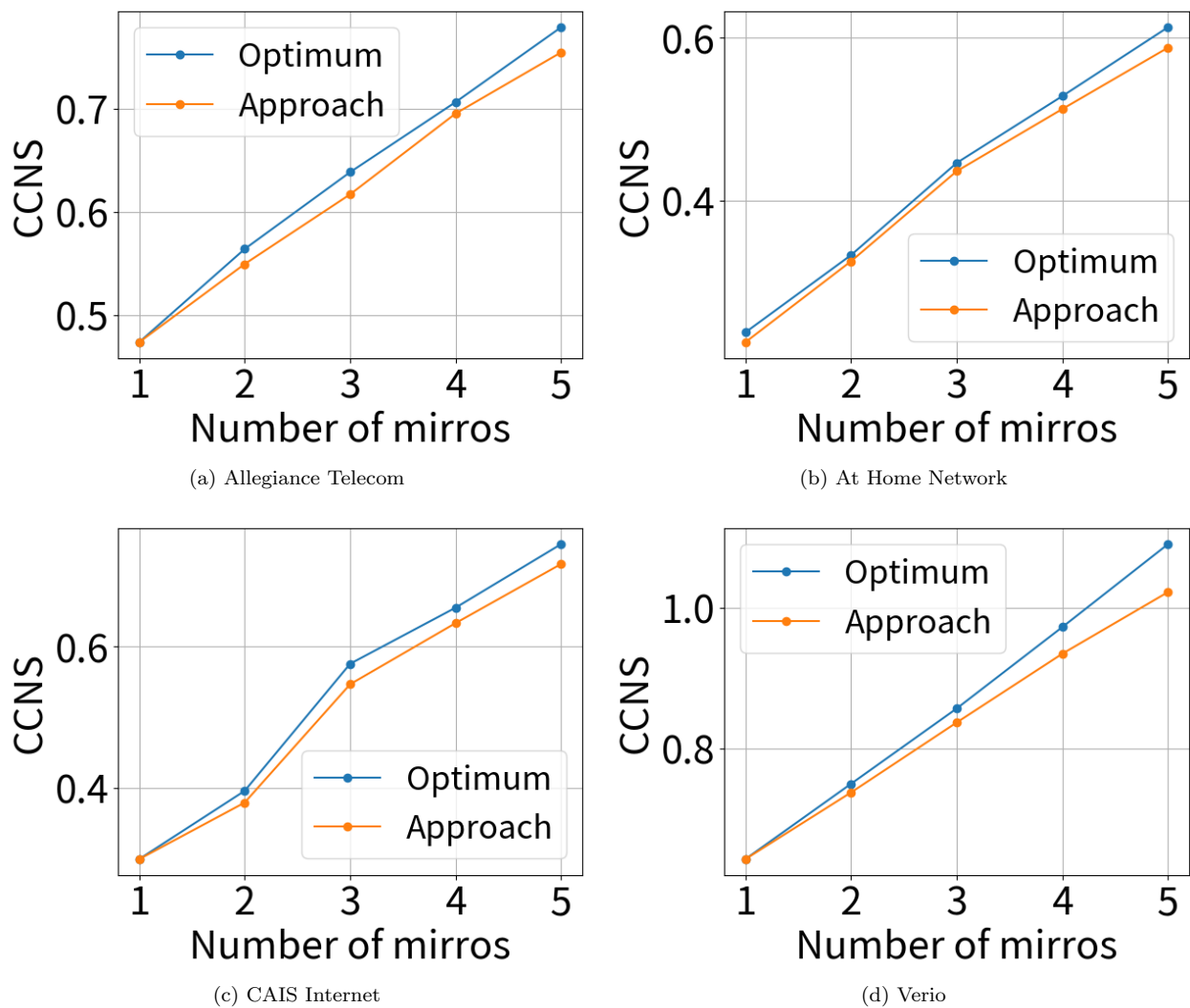


図 4.10: CCNS against mirror count

図 4.10 から、提案手法で選出したノードセットの近接中心性は最適解に近いということが分かった。これで提案手法の有効性を確認した。

4.4 動的なミラー生成法の性能

3.3 節では、固定的な周期でコンテンツの人気コンテンツを検知する方式の問題点を解決するための動的ミラー生成法を提案した。本節では、その動的ミラー生成法を用いることで、どのくらいの性能を向上できるかを評価する。なお、本節の評価結果は全て At Home Network というトポロジ上でシミュレーションを行って得た結果である。

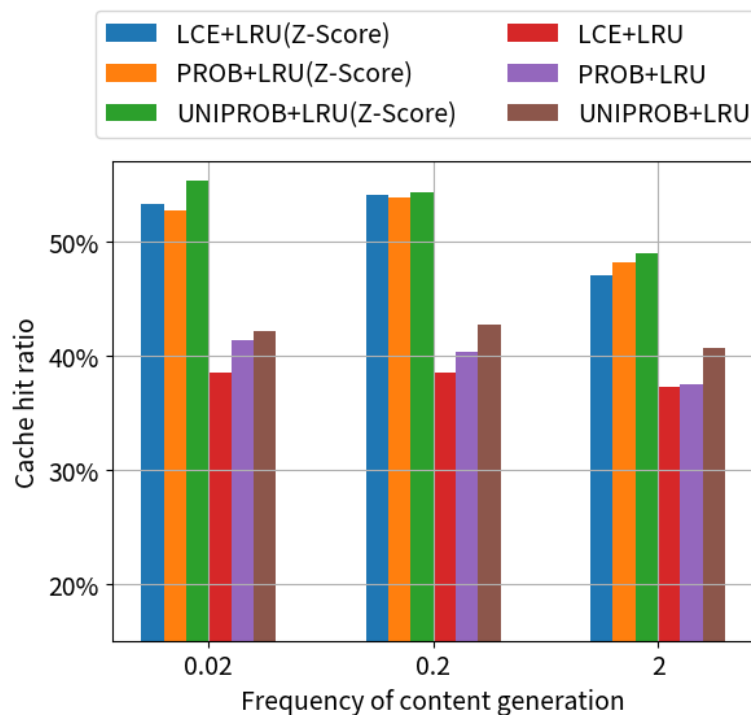


図 4.11: Cache hit rate against frequency of content generation

まず、異なるコンテンツ生成頻度におけるキャッシュヒット率を評価する。図 4.11 と図 4.12 から、Z スコア法に基づく動的なミラー生成法を導入した後のキャッシュ性能が導入しない時より高いことが分かった。このような結果を得た要因として考えられるのは、動的な人気度検知により、ミラーにリアルタイムの人気度変化を反映できたことである。

次はミラーサイズ比に対する提案手法を性能がどのように変化するかについて評価する。なお、4.2.3 節の評価結果から、特別な要求がない限り、ミラー数を 1 にするとときの性能が一番高いということが分かった。そのため、以降の結果は全てミラー数を 1 に設定した状態でシミュレーションを行って得た結果となる。

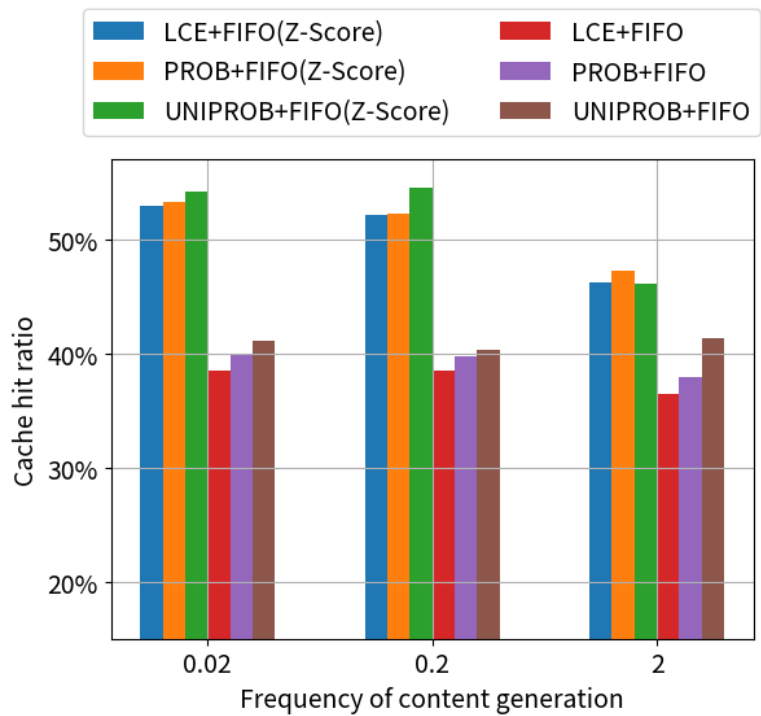


図 4.12: Cache hit rate against frequency of content generation

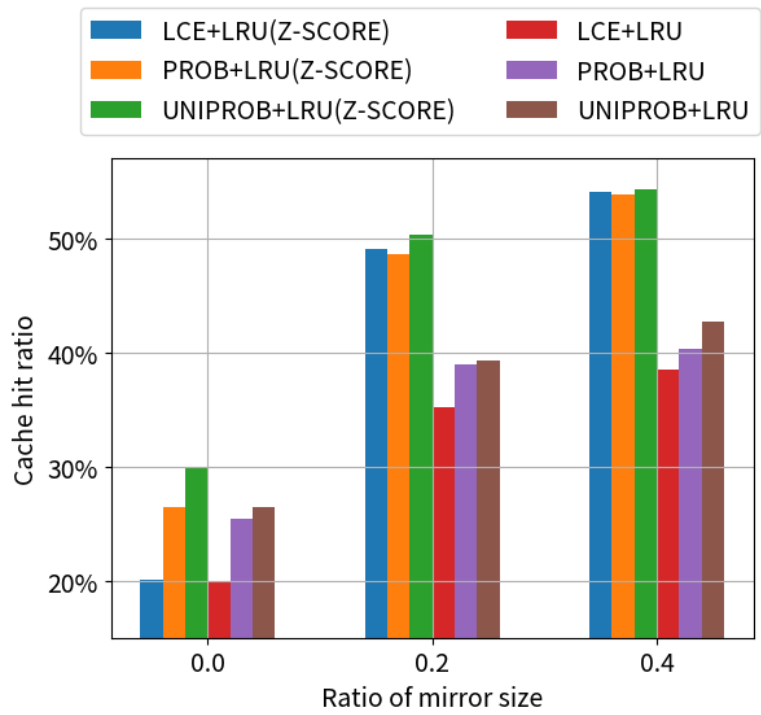


図 4.13: Cache hit rate against ratio of mirror size

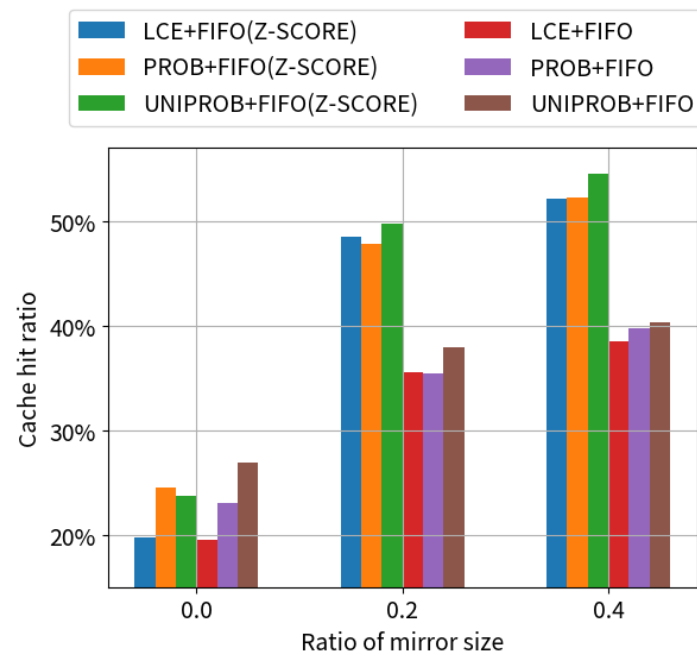


図 4.14: Cache hit rate against ratio of mirror size

図 4.13 と図 4.14 はそれぞれ置換ポリシーを LRU と FIFO と設定したときのミラーサイズに対するキャッシュヒット率を示したグラフである。この二つのグラフから、動的なミラー生成法を導入することにより、提案手法の性能を更に向上したことがわかった。また、動的なミラー生成法を導入した後、ミラーサイズ比が 0.2 と設定されるときは導入前にミラーサイズ比が 0.4 と設定されるときに性能より優れており、より小さいキャッシュ容量でより高い性能を発揮していることが分かった。

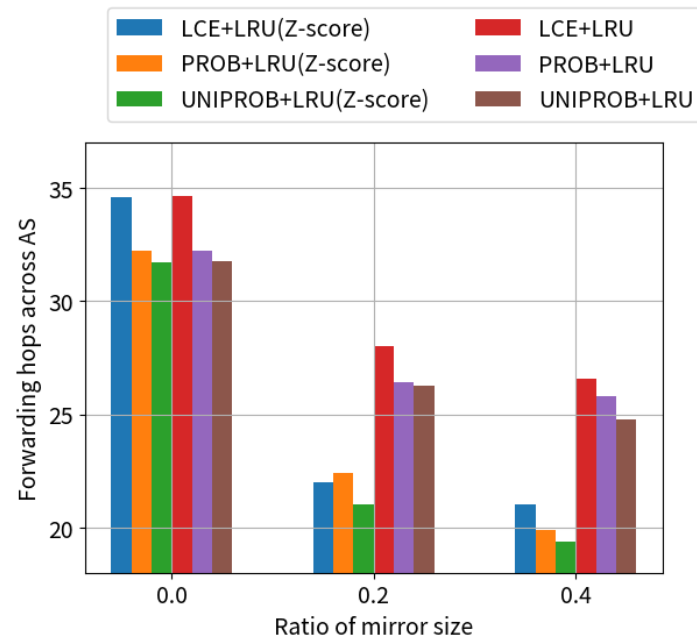


図 4.15: Average inter-AS hop length against frequency of content generation

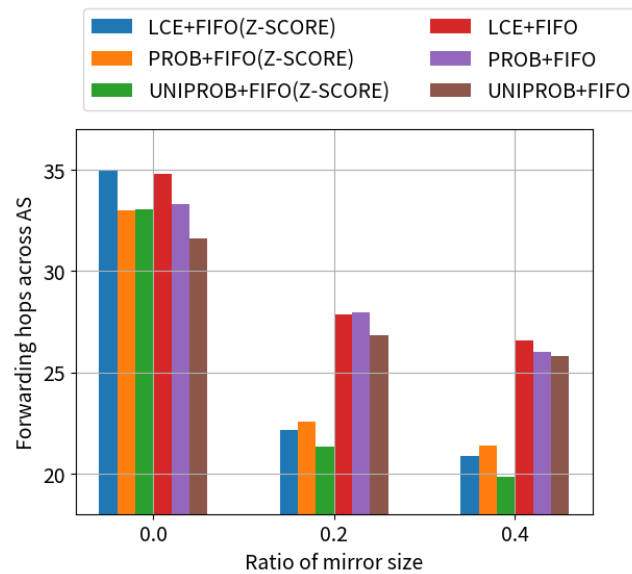


図 4.16: Average inter-AS hop length against frequency of content generation

同様に、図 4.15 と図 4.16 から、AS 間トラフィック量の抑制に関しても動的なミラー生成法のほうがより優れている性能をあらわす。これで、Z スコア法に基づく動的なミラー生成法はコンテンツ人気度に対して、より繊細な検知が可能であり、提案手法を性能を向上できると確認した。

第5章 結論

NDN ではルータにキャッシュを導入し、ルータのキャッシュからパブリッシャの代わりにコンテンツを配信する。しかし NDN のキャッシュ発見は Interest パケットの転送経路上でのみ実施されるため、NDN のキャッシュ性能の向上が困難である。そのため他の AS にパブリッシャが存在する場合、コストの高い AS 間トラフィック量の低減が困難である。

そこで本稿では、AS 内での高いキャッシュヒット率を実現し、AS 間トラフィック量の削減を低減することを目指す。NDN における動的ミラー構成法を提案し、AS 外部のコンテンツを AS 内部でミラー化し、AS 内部のトラフィックをミラーへ誘導することを提案した。また計算機シミュレーションにより、提案方式により AS 内のキャッシュヒット率が向上することを確認した。今後はとして案手法の実装及び実環境による性能やオーバーヘッドの評価を行う予定が挙げられる。

謝辞

本研究を行うに当たり，ご指導を頂いた上山教授に感謝します．また日々有益な議論をしていただいた先進ネットワーク研究室の皆様に感謝します．

参考文献

- [1] The caida homepage. <http://www.caida.org>. [Online].
- [2] The verio homepage. <http://www.verio.net>. [Online].
- [3] Vaibhav Aggarwal, Vaibhav Gupta, Prayag Singh, Kiran Sharma, and Neetu Sharma. Detection of spatial outlier by using improved z-score test. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 788–790, 2019.
- [4] Jianxun Cao, Dan Pei, Xiaoping Zhang, Beichuan Zhang, and Youjian Zhao. Fetching popular data from the nearest replica in ndn. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9. IEEE, 2016.
- [5] Jiachen Chen, Mayutan Arumaithurai, Lei Jiao, Xiaoming Fu, and KK Ramakrishnan. Copss: An efficient content oriented publish/subscribe system. In *2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*, pp. 99–110. IEEE, 2011.
- [6] Seyed Kaveh Fayazbakhsh, Yin Lin, Amin Tootoonchian, Ali Ghodsi, Teemu Koponen, Bruce Maggs, KC Ng, Vyas Sekar, and Scott Shenker. Less pain, most of the gain: Incrementally deployable icn. *ACM SIGCOMM Computer Communication Review*, Vol. 43, No. 4, pp. 147–158, 2013.
- [7] Chavoosh Ghasemi, Hamed Yousefi, and Beichuan Zhang. icdn: An ndn-based cdn. In *Proceedings of the 7th ACM Conference on Information-Centric Networking*, pp. 99–105, 2020.
- [8] Mohamed Hefeeda and Behrooz Noorizadeh. On the benefits of cooperative proxy caching for peer-to-peer traffic. *IEEE transactions on Parallel and Distributed Systems*, Vol. 21, No. 7, pp. 998–1010, 2009.
- [9] AKM Mahmudul Hoque, Syed Obaid Amin, Adam Alyyan, Beichuan Zhang, Lixia Zhang, and Lan Wang. Nlsr: Named-data link state routing protocol. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pp. 15–20, 2013.
- [10] Yuanzhi Kan, Quan Zheng, Jian Yang, and Xiaobin Tan. A cache invalidation strategy based on publish/subscribe for named data networking. *IEEE Access*, Vol. 8, pp. 80074–80085, 2020.
- [11] Dirk Kutscher, Suyong Eum, Kostas Pentikousis, Ioannis Psaras, Daniel Corujo, Damien Saucez, Thomas C Schmidt, and Matthias Waehlich. Information-centric networking (icn) research challenges. *Internet Research Task Force (IRTF)*, pp. 1–38, 2016.
- [12] Jun Li, Hao Wu, Bin Liu, Jianyuan Lu, Yi Wang, Xin Wang, Yanyong Zhang, and Lijun Dong. Popularity-driven coordinated caching in named data networking. In *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, pp. 15–26, 2012.

-
- [13] Ge Ma, Zhen Chen, Junwei Cao, Zhenhua Guo, Yixin Jiang, and Xiaobin Guo. A tentative comparison on cdn and ndn. In *2014 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 2893–2898. IEEE, 2014.
- [14] Rajesh Nishtala, Hans Fugal, Steven Grimm, Marc Kwiatkowski, Herman Lee, Harry C Li, Ryan McElroy, Mike Paleczny, Daniel Peek, Paul Saab, et al. Scaling memcache at facebook. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pp. 385–398, 2013.
- [15] Boubakr Nour, Kashif Sharif, Fan Li, Song Yang, Hassine Mounsla, and Yu Wang. Icn publisher-subscriber models: Challenges and group-based communication. *IEEE Network*, Vol. 33, No. 6, pp. 156–163, 2019.
- [16] Haitao Wang, Sripriya Adhatarao, Mayutan Arumaithurai, and Xiaoming Fu. Copss-lite: Lightweight icn based pub/sub for iot environments. *arXiv preprint arXiv:1706.03695*, 2017.
- [17] Guoqiang Zhang, Yang Li, and Tao Lin. Caching in information centric networking: A survey. *Computer networks*, Vol. 57, No. 16, pp. 3128–3141, 2013.
- [18] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. Named data networking. *ACM SIGCOMM Computer Communication Review*, Vol. 44, No. 3, pp. 66–73, 2014.
- [19] Meng Zhang, Hongbin Luo, and Hongke Zhang. A survey of caching mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 3, pp. 1473–1499, 2015.

学会発表リスト

- [1] Cao Xuheng, 上山 憲昭, “NDN における AS 間トラフィック削減のための動的ミラー構成法“, 電子情報通信学会 2023 年総合大会, B-14-8, 大宮/オンライン, 2023 年 3 月
- [2] Cao Xuheng, 上山 憲昭, “NDN の AS 間トラフィック削減のためのミラー動的配置法“, 電子情報通信学会 情報通信マネジメント (ICM) 研究会, ICM2023-16, 函館/オンライン, 2023 年 7 月
- [3] Cao Xuheng, 上山 憲昭, “NDN の AS 間トラフィック削減を目的とした動的ミラー配置法“, 電子情報通信学会 2023 年ソサイエティ大会, BS-4-8, 名古屋, 2023 年 9 月
- [4] Xuheng Cao, Noriaki Kamiyama, ”Dynamic Mirror Placement for Reducing Inter-AS Traffic in NDN”, IEEE CCNC2024 (Work in progress session), 2024