

令和5年度 春学期 卒業研究3 (BI)  
学士論文

題目 NDNの移動コストを考慮した  
オリジナル配置制御

指導教員 上山憲昭 教授

立命館大学 情報理工学部 セキュリティ・ネットワークコース

学籍番号 2600190169-8

品田奈美

令和5年8月2日

## 概要

デジタルコンテンツを効率的に配信することができるネットワークアーキテクチャとして、情報指向ネットワーク (ICN: information-centric networking) が大きな注目を集めている。現在のネットワークは IP ネットワークだが、ICN では IP アドレスを用いずにコンテンツ名で配信要求を行う。IP アドレス数 (ホスト数) とコンテンツ数を比較した場合、コンテンツ数は膨大である。それに加えて、コンテンツ名は場所に依存しないため、ICN ルータの転送テーブル (FIB: forwarding information base) のエントリ集約は困難である。コンテンツ名が同一の文字列を含む場合はエントリ集約が可能になるため、似た名称のコンテンツが同一ノードに存在するとエントリを効率的に集約できる。また、現在のインターネットでは、コンテンツを効率的に配信する技術として CDN (content delivery network) が広く利用されているが、ユーザの配信品質向上とトラフィック量の削減という CDN の目的は ICN によって達成される。

本稿では、CDN を ICN のオリジナル提供プラットフォームと位置付け、コンテンツの移動コストを考慮した上で、ICN の FIB エントリを効果的に削減する手法を提案する。オリジナルが配置されているノードから再配置するノードまでのホップ数を移動コストとし、それに上限を設けることで総移動コストを削減する。さらに移動コストの上限をもとにコンテンツを配置するノードの優先度ランキングを作成し、そのランキングの上位のノードからコンテンツの再配置を行う。再配置するランクを制限し、平均 FIB サイズと総移動コストが最適になる配置制限ランクを示す。

# 目次

|                                |           |
|--------------------------------|-----------|
| 概要                             | 1         |
| <b>第1章 序論</b>                  | <b>3</b>  |
| 1.1 研究の背景                      | 3         |
| 1.2 研究の目的                      | 4         |
| <b>第2章 関連研究</b>                | <b>5</b>  |
| 2.1 LPMを用いたFIB集約法              | 5         |
| 2.2 CDNを用いたコンテンツ配置法            | 6         |
| <b>第3章 移動コストを考慮したコンテンツ配置設計</b> | <b>7</b>  |
| 3.1 コンテンツ配置ノードの優先度ランキングの設計     | 7         |
| 3.2 優先度ランキングを用いたコンテンツ配置アルゴリズム  | 8         |
| <b>第4章 シミュレーション評価</b>          | <b>11</b> |
| 4.1 シミュレーション条件                 | 11        |
| 4.2 コンテンツ配置ノードの優先度ランキング        | 12        |
| 4.3 シミュレーションの結果と考察             | 14        |
| <b>第5章 まとめ</b>                 | <b>18</b> |
| 謝辞                             | 19        |

# 第1章 序論

## 1.1 研究の背景

インターネットや SNS の普及によって増大した YouTube などのユーザ生成コンテンツ (UGC) や、映画やドラマなどのリッチコンテンツの配信によって生成されるトラフィックは、インターネット上のトラフィックの大部分を占めている。さらに、高速・大容量・同時多接続の 5G 通信の普及に伴い、IoT (internet of things) は農業や企業の工場、スマートホーム、スマートシティなどで実用的になりつつあり、IoT デバイスから取得した膨大なセンサデータを制御する処理が必要である。そこで、動画コンテンツや IoT のセンサデータ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 ICN (information-centric networking) が注目されている [8]。ICN ではルータにコンテンツをキャッシュし、コンテンツ名を使用してパケットをルーティングするため、コンテンツの配信元を特定せずに直接コンテンツを要求することが可能である。そのため、膨大な数のデータが生成される IoT 通信や、コンテンツの配信元が事前にわからない災害時の通信において、効率的にデータ通信を行うことができるネットワークとして注目されている。ICN の実現のために提案されているネットワークは、TRAND[12] やコンテンツ中心型ネットワーク (CCN: content-centric networking)[13]、データ指向ネットワークアーキテクチャ (DONA: data-oriented network architecture)[14]、名前付きデータネットワーク (NDN: named data networking) [16] などがある [15]。本稿では ICN のアーキテクチャとして NDN を想定する。

NDN ではコンテンツアイテムを要求するパケットは Interest と呼ばれている。従来の IP アドレスを用いた通信方法と同様に、FIB (forwarding information base) と呼ばれる経路制御表を参照することで NH (next hop) に Interest を転送する。FIB では Prefix と NH の組によってエントリが作成される。IP ルータの FIB では転送先の IP アドレス、NDN ルータの FIB ではコンテンツ名が Prefix として用いられ、コンテンツ名は Prefix とコンテンツ ID で構成される。例えば、“ritsumei-u/sample.pdf” の場合、“ritsumei-u” が Prefix、“sample.pdf” はコンテンツ ID となる。NDN のコンテンツのオリジナルは Publisher のホストに存在し、Publisher はコンテンツ名の Prefix を隣接するルータに広告する [2]。Prefix 広告を受信する NDN ルータは、ユーザである Subscriber から送信された Interest が、要求されたコンテンツによって Publisher のホストに転送されるように FIB を構成する。

FIB に全ての Prefix のエントリを作成した場合、FIB に必要なメモリサイズは大幅に増加する。したがって、FIB のメモリコストを削減するためには、FIB エントリの集約が必要である。IP 通信の場合、ネットワークプロバイダは割り当てられたアドレスブロックの一部を顧客組織に割り当てるため、IP アドレスには地理的な局所性があり、IP アドレスは階層構造になっている。しかし、NDN の場合、同じ組織のコンテンツを提供する Publisher のホストは同じ地域に存在する傾向はあるが、組織の名前に地理的な局所性は

ない。そのため、IP 通信の Prefix 集約と比較して、NDN の FIB における Prefix 集約は困難になる。例えば、Web ページをコンテンツと見なした場合、約  $10^{11}$  のコンテンツ名があり、同じ組織の Prefix を 1 つの Prefix に集約したとしても、約  $10^9$  のコンテンツ名の Prefix がある [4]。異なる組織間の Prefix 集約は難しく、NDN の各 FIB には約  $10^9$  エントリが必要である [4] が、IP 通信の各 FIB に必要なのは約  $10^5$  エントリである。ハッシュ表 [5] に基づいたエントリ検索を用いる場合、数百万の Prefix に対して数メガバイトのメモリが FIB に必要であるため、SRAM を使用した FIB の実装は困難である。

インターネットでは、効率的なコンテンツ配信の仕組みとして CDN (content delivery network) が広く使用されている。CDN では、オリジンサーバからコンテンツのコピーを取得したキャッシュサーバがコンテンツ配信を行うことで、距離が近いサーバからコンテンツが取得できるため、配信品質の向上とネットワークトラフィック量の削減が実現可能である。NDN では、ユーザから近いルータにコンテンツがキャッシュされ、配信される可能性が高いことから、CDN の目的である配信品質の向上とネットワークトラフィックの削減は NDN によって満たされる。そこで先行研究では、CDN のキャッシュサーバはキャッシュされたコンテンツの Prefix をネットワークに広告可能であること [6] や、Publisher のホストになることが可能であることから、コンテンツのコピーを CDN キャッシュサーバに配置することで、NDN ルータの FIB エントリ集約を行った [1]。

しかし [1] ではノードの位置は考慮されていない。ノード位置を考慮しないと発生する可能性がある問題として、トラフィックの負荷集中が考えられる。また、オリジナル位置から離れた場所にあるノードが再配置場所であるコンテンツが多い場合、コンテンツの移動に要するトラフィック量が増加する。これらの問題に対処するために、コンテンツを再配置するノードを考慮する必要があると考える。

## 1.2 研究の目的

本稿では [1] を拡張し、ノードの位置を考慮する。コンテンツのオリジナルが配置されているノードから再配置するノードまでのホップ数を移動コストと定義し、コンテンツを配置した際の移動コストについて考慮する。移動コストを考慮することで、コンテンツの再配置に要するトラフィック量の削減が期待される。[1] のコンテンツ配置アルゴリズムに、制限ホップ数以内のノードのみ配置する条件を追加することで移動コストを削減する。また、移動コストの制限によって平均 FIB サイズに与える影響を明らかにし、総移動コストと平均 FIB サイズが最適になるコンテンツ配置方法を提案する。

## 第2章 関連研究

ICN ルータの FIB エントリ数を削減する既存の方法として、(1) 部分キャッシュ、(2) ルート集約、(3) フラッディング、(4) ブルームフィルタ、(5) LPM が提案されている。部分キャッシュでは、すべての Prefix ではなく、一部の Prefix に対して FIB エントリが作成される [3][4]。[3] では DNS を使用し、エントリがルータの FIB に存在しない名前を解決する方法を提案している。到着する Interest の Prefix エントリがルータの FIB に存在しない場合、ルータは否定応答を Subscriber に返す。そして Subscriber は同じドメインで DNS によるルータの FIB エントリが存在する Prefix を得て、その Prefix に対し Interest を送信する。また [4] では、ルックアンドキャッシュアプローチを提案している。ルータは NRS (name routing system) サーバから取得したルーティング情報をルータの FIB にキャッシュする。このアプローチでは FIB のサイズは縮小されるが、名前検索が必要となる。

2 番目のアプローチであるルート集約では、すべての Interest が NAC (name collector) と呼ばれる同一のルータを通過するように FIB が設定される [11]。NAC がルートノードであるツリートポロジで NAC に向けて Interest が送信されるため、ルータに必要な FIB サイズが削減されるが、このアプローチでは Interest の転送ホップ長が増加する。

3 番目のアプローチのフラッディングでは、FIB 検索を行わずにルータの全ての出力面に Interest がブロードキャストされる [6][7]。[6] ではルータで Interest をブロードキャストし、Prefix をキャッシュしている隣接ルータは要求元ルータにルータ情報を返す。[7] では、低人気なコンテンツアイテムの Interest を転送し、FIB を使わずに高人気のコンテンツアイテムの Interest をブロードキャストする方法を提案している。人気コンテンツのコピーは多くのルータでキャッシュされる可能性が高いため、Interest をブロードキャストすることで、要求されたコンテンツを持つルータに Interest が高い確率で届く。ただし、Interest は冗長に転送されるため、ネットワークリンクが過負荷になる可能性がある。

4 番目のアプローチであるブルームフィルタでは、ルータが各出力面で提供されるブルームフィルタを使用し、到着する各 Interest を各出力面に転送するか否かを判断する [9][10]。ブルームフィルタを使用することで、ルータは限られた数のメモリアクセスで小さなサイズのメモリを使用して、Interest 転送の決定ができる。ただし、Interest を誤った出力面に転送する可能性があり、冗長な Interest 送信によりネットワーク負荷が増加する。

### 2.1 LPM を用いた FIB 集約法

[1] では、IP ルータで FIB サイズの削減するために用いる LPM (longest prefix matching) を NDN ルータに適用している。アプローチとして、(1) URL の文字の単位、(2) URL のコンポーネントのユニット、以上2つがあるが、ここでは (2) を利用している。コンポーネントは URL のピリオドで区切られた文字列として定義している。例えば、“www.ritsumei.ac.jp”

の URL は “www”, “ritsumei”, “ac”, “jp” で構成される 4 つのコンポーネントである。NDN の Prefix は分離したコンポーネントを逆順に並び替えて連結したもので、つまり、“jp/ac/ritsumei/www” である。URL の最初のコンポーネントは TLD (top level domain) であり、URL の 2 番目のコンポーネントは SLD (second level domain) とし、URL のコンポーネントは LPM の単位として使用している。

各ノード  $n$  にて、ノード  $n$  にオリジナルが割り当てられたオブジェクトを除く全てのオブジェクトに対して FIB エントリを作成する。このアルゴリズムは各ノードで個別に実行できるため、1 つのノードに焦点を当てる。FIB エントリ  $x$  の Prefix を  $P(x)$  で、 $P(x)$  の最初から  $s$  番目までのコンポーネントを  $p(x, s)$  で示す。さらに、ルーティングアルゴリズムによって設定される FIB エントリ  $x$  の出力ポートを  $F(x)$  とする。また、 $M$  をまだチェックされていない FIB エントリの集合、 $C$  を URL のコンポーネントの最大数と定義する。

最初に TLD を対象に FIB エントリ、つまり  $s = 1$  を集約し、次のステップでは SLD までの FIB エントリ  $s = 2$  を集約する。すべてのコンポーネントの位置  $1, 2, \dots, C - 1$  で集約が完了するまで、この手順が繰り返される。この方法は FIB エントリを逆方向、つまり  $s = C - 1$  から  $s = 1$  の順に集約することもできる。 $s = 1$  から  $C - 1$  の順で集約する手順を短期集約、逆方向の  $s = C - 1$  から  $1$  の順を長期集約としている。

Web オブジェクトのオリジナル位置で、短期集約、長期集約のアルゴリズムを適用した結果、FIB サイズを約 20~50 %削減できることを示している。また、2 つの集約方法の効果の差については、集約に必要な計算時間が、長期集約より短期集約の方がはるかに短いため、短期集約の方が優れた効果を発揮することを示している。そして、TLD と SLD で集約すると FIB エントリのほぼ全ての集約効果が得られることを確認している。

## 2.2 CDN を用いたコンテンツ配置法

[1] では、TLD や SLD のみに着目し、該当オブジェクトの上位から順にノードの位置を考慮せずにコンテンツ配置を行っている。さらに、負荷集中を避けるため、各ノードにオブジェクトの配置上限数を設定することで、全ノードに可能な限り均等にオブジェクトを配置している。FIB エントリは TLD または SLD と NH が共通しているエントリを集約できるため、同一 TLD または SLD を同一ノードに配置することにより、FIB 集約の効果を向上させることができる。さらに FIB 集約の効果向上のために、各 TLD および SLD を使用した URL の数を調査し、その数が多い TLD の Web オブジェクトを同一ノードに配置した。この方式によって、コンテンツをオリジナルの位置に配置したときと比較して、FIB サイズを約 45 %削減できることを示している。

## 第3章 移動コストを考慮したコンテンツ配置設計

既存研究 [1] ではノードの位置を考慮せず，Web オブジェクト数の降順にコンテンツの配置を行った．そこで本節では，オリジナルの配置場所から再配置場所までのホップ数に上限値を設定し，ホップ数が上限値以下のノードにコンテンツを配置する方式を提案する．

### 3.1 コンテンツ配置ノードの優先度ランキングの設計

コンテンツのオリジナルが配置されているノードから再配置するノードまでのホップ数を移動コストとする．移動コストの制限をもとに，TLD 別に配置できるコンテンツ数をノードごとに調査し，コンテンツ配置ノードの優先度ランキングを作成する．

図 3.1 にコンテンツ配置ノードの優先度ランキングの作成手順を示す．まず，各ノードに対して，移動コストの制限以内であるノードを調査する．同一 TLD のコンテンツで，配置可能なコンテンツ数をノードごとに集計し，その集計をもとに降順で配置ノードの優先度ランキングを作成する．この時，配置可能なコンテンツが 0 個のノードはランキング外にする．例えば，TLD が “com” のコンテンツの配置可能ノードが図 3.1 の (1) のようになっているとする．ノード 1 に配置可能なのは，“com/ab”，“com/ef”，“com/mn” の 3 つのコンテンツである．同様にノード 2 について調査すると 5 つすべてのコンテンツが配置可能であることがわかる．これをすべてのノードに対して行った結果が (2) である．集計した (2) の結果をもとに降順でソートをすると (3) のような配置ノードの優先度ランキングが作成できる．

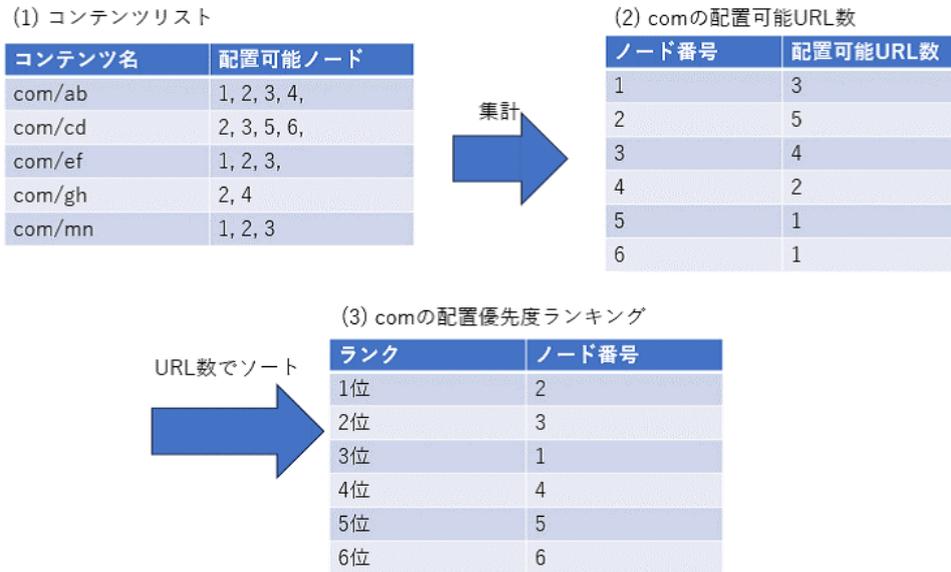


図 3.1: 配置ノードの優先度ランキング作成の手順

### 3.2 優先度ランキングを用いたコンテンツ配置アルゴリズム

既存研究 [1] のコンテンツ配置アルゴリズムに配置ノードの優先度ランキングを追加する。トラフィック量を緩和するために、1つのノードに配置される Web オブジェクトの最大数を上限値  $B$  以下に制限する。  $B$  は、ノードの数  $N$  で割った Web オブジェクトの数  $W$  以上でなければならない。このアルゴリズムではネットワークのすべてのノードに CDN キャッシュサーバが提供されていることを想定している。各 TLD を使用して、コンテンツの数の降順で TLD をソートし、  $D_1(x)$  を  $x$  番目にランク付けされた TLD と定義する。また、各 SLD を使用して、TLD と同様にコンテンツの数の降順に各 TLD で SLD をソートし、  $D_2(s, y)$  を  $y$  番目にランク付けされた SLD、  $s$  を TLD と定義する。  $M_1$  を異なる TLD の数、  $M_2(s)$  を TLD が  $s$  である異なる SLD の数として定義する。さらに、  $U_1(x)$  は TLD として  $D_1(x)$  を持つコンテンツの集合を示し、  $U_2(s, y)$  は TLD が  $s$ 、 SLD が  $D_2(s, y)$  を有するコンテンツの集合を示す。  $m_1(x)$  と  $m_2(s, y)$  を、  $U_1(x)$  と  $U_2(s, y)$  に含まれるコンテンツの数として定義する。  $A_n$  はノード  $n$  に配置できるコンテンツの数を示し、  $n^*$  は  $A_n$  の最大値を持つノードを示す。  $r(x, t)$  は  $D_1(x)$  の配置ノードの優先度ランキングの  $t$  位のノードを示し、ランキングの最下位を  $t^*$  とする。そして、優先度ランキングの  $T$  位までコンテンツの配置を行う。次に FIB サイズを抑えるためにノードにコンテンツを配置するアルゴリズムを図 3.2 に示す。

---

```

1:  Initializes  $x = 1$  and  $A_n = B$  for all  $n$ 
2:  while  $x \leq M_1$  do
3:      Finds  $n^*$ 
4:      if  $m_1(x) \leq A_{n^*}$  then
5:          Initializes  $t = 1$ 
6:          while  $t \leq t^*$  and  $t \leq T$  do
7:              if  $n = r(x, t)$  then
8:                  Places  $U_1(x)$  at node  $n^*$  and subtracts  $m_1(x)$  from  $A_{n^*}$ 
9:                  break while
10:             else
11:                 Increments  $t$ 
12:             end if
13:         end while
14:     else
15:         Initializes  $y = 1$ 
16:         while  $y \leq M_2(D_1(x))$  do
17:             while  $m_2(D_1(x), y) > A_{n^*}$  do
18:                 Initializes  $t = 1$ 
19:                 while  $t \leq t^*$  and  $t \leq T$  do
20:                     if  $n = r(x, t)$  then
21:                         Places  $U_2(D_1(x), y)$  at node  $n^*$  and subtracts  $m_2(D_1(x), y)$  from  $A_{n^*}$ 
22:                         break while
23:                     else
24:                         Increments  $t$ 
25:                     end if
26:                 end while
27:                 Increments  $y$  and finds  $n^*$ 
28:             end while
29:         end while
30:     end if
31:     Increments  $x$ 
32: end while

```

---

図 3.2: 優先度ランキングを用いたコンテンツ配置アルゴリズム

コンテンツ再配置の対象を TLD のリストの上位から順に選択する。配置できるコンテンツの数が最大のノード  $n^*$  を再配置するノードとして選択する。  $m_1(x)$  の降順に参照し、  $m_1(x) \leq A_{n^*}$  の場合、かつ配置ノードの優先度ランキングを上位から最下位まで参照し、ランキングに含まれている、かつ  $T$  位以内であれば、ノード  $n^*$  に  $U_1(x)$  を割り当てる。  $m_1(x) > A_{n^*}$  の場合、  $U_1(x)$  を  $y = 1, 2, \dots, M_2(D_1(x))$  として、  $U_2(D_1(x), y)$  に分割する。 TLD の時と同様に、配置ノードの優先度ランキング上位から最下位まで、もしくは  $T$  位まで参照し、ランキングに含まれていれば、  $U_2(D_1(x), y)$  を各  $y$  のノード  $n^*$  へ割り当てる。その結果、  $u_n$  がノード  $n$  に配置された URL の数として、  $u_n \leq B$  を満たしつつ、各ノードで FIB エントリの集約効果が向上することが期待できる。さらに、コンテンツ配置ノードの優先度ランキングに含まれているノードにのみ割り当てることで、移動コストの削減を実現することができる。コンテンツ配置ノードの優先度ランキングの下位まで参照することによって、FIB 集約の効果が期待できるが、一方で移動コストの増加が懸念される。平均 FIB サイズと総移動コストを最適なバランスにするために、コンテンツ再配置の対象にする優先度ランキングの順位を  $T$  位までに制限する。

## 第4章 シミュレーション評価

### 4.1 シミュレーション条件

Alexa[19]のWebページで公開されているアクセス数のランキングから、16の各カテゴリに対して上位500の合計8,000のWebページのURLを収集した。ここからシミュレーションで扱うデータを厳選していく。一部のWebブラウザには、Webページにアクセスする際にさまざまな通信プロパティを含むHAR (HTTP Archive) ファイルをエクスポートする機能がある。それを使い、収集した各URLに対し、Webオブジェクトの配信サーバのURLや取得までの遅延時間などの情報を取得すると、一部のWebページでタイムアウトエラーなどの問題が発生した。HAR ファイルを正常に取得できたのは、8,000のうち7,604のWebページだった。7,604のWebページに含まれるWebオブジェクトの累積数は679,380であり、これらのオブジェクトのURLをHARファイルから抽出する。次に、オリジナルの場所にあるコンテンツのみにする。CDNが利用されている場合、多くのオブジェクトはオリジンサーバではなく、キャッシュサーバから配信されている。この場合、DNSクエリはCDNプロバイダのDNSサーバにリダイレクトされるため、DNSレコードに“CNAME”の文字列が含まれているものはCDNから配信されたものである可能性が高い。抽出した各URLにdigコマンドを使用してDNSレコードを取得し、“CNAME”が含まれるURLを除外した。さらに、CDNに関連する5つのキーワード (cdn, edge, akam, cloudfront, cloudflare) が含まれるURLも除外し、679,380のURLから18,547のURLが残った。そして、MaxMind[20]が提供しているGeoIP APIを使用し、各Webオブジェクトの送信元の国名、都市名、および座標を取得した。18,547のURLのうち約64%がアメリカ合衆国から配信されていることを確認し、評価に使用するものをアメリカ合衆国にオリジナルが存在する12,010のWebオブジェクトのURLと座標に限定した。

数値評価では、アメリカ合衆国のバックボーンネットワークであるInternet2のトポロジを使用した[21]。Layer3 serviceを提供する12ノードを使用し、図4.1に選択した12ノードで構成されるInternet2のトポロジを示す。

各ノード間のホップ数を調査し、最も遠いノード間のホップ数が4であるため、移動コストの上限を1, 2でそれぞれシミュレーションを行った。

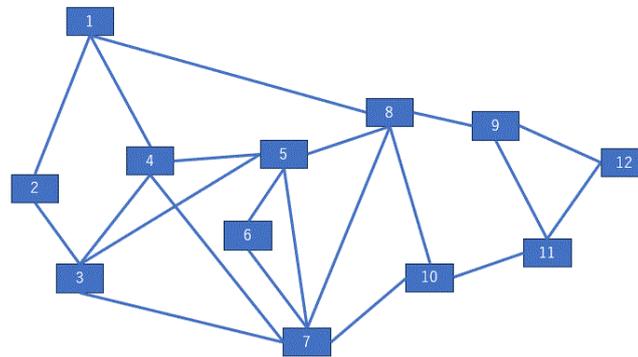


図 4.1: Internet2

## 4.2 コンテンツ配置ノードの優先度ランキング

各ノードについて、移動コストの上限を 2 に設定した時の各ノードに対するコンテンツ配置可能なノード番号を表 4.1 に示す。これに基づいて、各ノードに配置可能なコンテンツ数を計測する。計測した数値をもとに、コンテンツ配置ノードの優先度ランキングを作成し、このランキングの上位のノードから再配置を行う。

アルゴリズムの動作確認、およびプログラムの実装に問題がないかを確認するために、すべてのデータの中から一部のデータを取り出して、コンテンツ配置ノードの優先度ランキングの構成の調査を行った。選択したデータとして、最もコンテンツ数が多い TLD である “com” を使用した。“com” のコンテンツ配置ノードの優先度ランキングから、ランクに対応するノード番号とそのノードに配置可能なコンテンツ数を表 4.2 に示す。配置可能なコンテンツが多いノードから降順にランキングが構成されていることを確認した。同一 TLD のコンテンツ数が少なく、オリジナル位置が局所的な場合は、この限りではないが、“com” は全データの中で最もコンテンツ数が多いため、オリジナルの位置がすべてのノードに散在していると考えられる。また、表 4.1 より、ノード 8 はすべてのノードから再配置を行うことができるため、コンテンツ配置ノードの優先度ランキングの上位になると推定される。アルゴリズムのシミュレーション結果は推定と矛盾していない。

表 4.1: 各ノードに対する配置可能ノード

| ノード     | 配置可能ノード                                |
|---------|--|
| Node 1  | 1, 2, 3, 4, 5, 7, 8, 9, 10,            |
| Node 2  | 1, 2, 3, 4, 5, 7, 8,                   |
| Node 3  | 1, 2, 3, 4, 5, 6, 7, 8, 10,            |
| Node 4  | 1, 2, 3, 4, 5, 6, 7, 8, 10,            |
| Node 5  | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,         |
| Node 6  | 3, 4, 5, 6, 7, 8, 10,                  |
| Node 7  | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,     |
| Node 8  | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, |
| Node 9  | 1, 5, 7, 8, 9, 10, 11, 12,             |
| Node 10 | 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,    |
| Node 11 | 7, 8, 9, 10, 11, 12,                   |
| Node 12 | 8, 9, 10, 11, 12,                      |

表 4.2: TLD が “com” のコンテンツ配置ノードの配置優先度ランキング

| Rank | Node | 配置可能コンテンツ数 |
|------|------|------------|
| 1    | 8    | 8292       |
| 2    | 7    | 7464       |
| 3    | 10   | 6671       |
| 4    | 5    | 6373       |
| 5    | 3    | 6213       |
| 6    | 4    | 6213       |
| 7    | 1    | 6078       |
| 8    | 9    | 5942       |
| 9    | 2    | 5694       |
| 10   | 6    | 3638       |
| 11   | 11   | 2785       |
| 12   | 12   | 2614       |

### 4.3 シミュレーションの結果と考察

コンテンツを配置する際に、配置ノードの優先度ランキングの順位に制限を設け、各制限ランク時の平均FIBサイズと総移動コストを計測した。移動コストの上限が1, 2の時の平均FIBサイズを図4.2, 図4.3に、移動コストの上限が1, 2の時の総移動コストを図4.4, 図4.5にそれぞれ示す。また、再配置されなかったコンテンツ数についても調査を行った。その結果を図4.6, 図4.7に示す。

配置制限ランクを拡大すると、平均FIBサイズは減少傾向になった。これは、制限ランクの拡大によって、再配置されるコンテンツが増加し、集約できるFIBエントリが増加したからだと考えられる。また、移動コストの上限が1の時はランキング3位まで、移動コストの上限が2の時はランキング6位までに制限すると、平均FIBサイズを大幅に削減できた。配置制限ランクが12, つまりランキングに含まれる全てのノードに配置を行った結果は既存研究[1]とほぼ変わらない結果になった。

総移動コストは、配置制限ランクを拡大すると、再配置されるコンテンツが増加するため、増加する傾向になった。また、移動コストの上限が1の時はランキング2位から3位、移動コストの上限が2の時はランキング5位から6位の総移動コストの増加度が大きくなった。配置制限ランクが12の時は平均FIBサイズと同様に、既存研究とほぼ変わらない結果になった。

再配置されなかったコンテンツ数については、移動コストの上限に関わらず、減少傾向になった。配置制限ランクを拡大すると、必然的に再配置の対象になるコンテンツが増えるため、それに比例して再配置されなかったコンテンツ数は減少する結果となった。この結果から、配置制限ランクの拡大によって再配置するコンテンツが増えていることがわかる。

移動コストの上限が1の時、配置制限ランクを3に設定すると、既存研究と比較して平均FIBサイズは約21%増加するが、総移動コストは約67%削減できた。移動コストの上限が2の時は、配置制限ランクを6に設定すると、平均FIBサイズは約13%増加するが、総移動コストは約45%削減できた。

シミュレーション結果から、最適な配置制限ランクは、移動コストの上限が1の時は3、移動コストの上限が2の時は6となる。このことから、全ノード間の最大移動コストに対する移動コストの上限の比率と、最大ランク数に対する配置制限ランクの比率を同じにすることで、平均FIBサイズと総移動コストが最適な値を示すと考えられる。

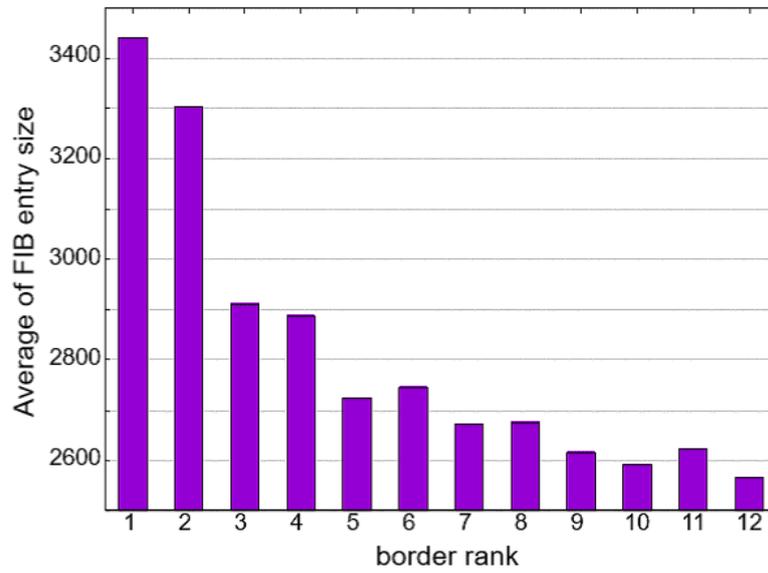


図 4.2: 上限移動コストが 1 のときの平均 FIB サイズ

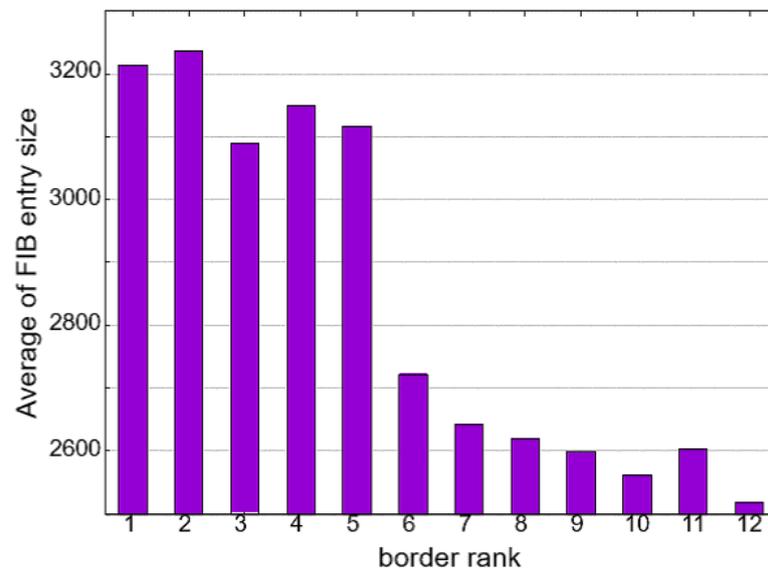


図 4.3: 上限移動コストが 2 のときの平均 FIB サイズ

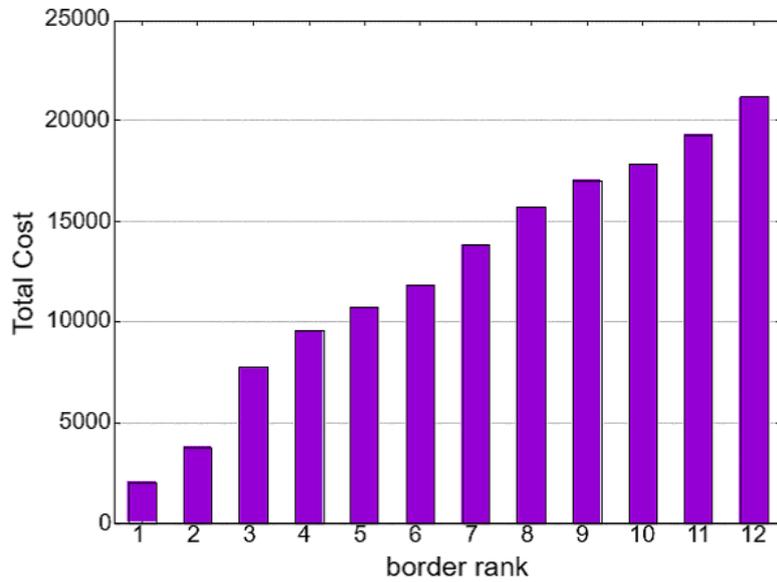


図 4.4: 上限移動コストが 1 のときの総移動コスト

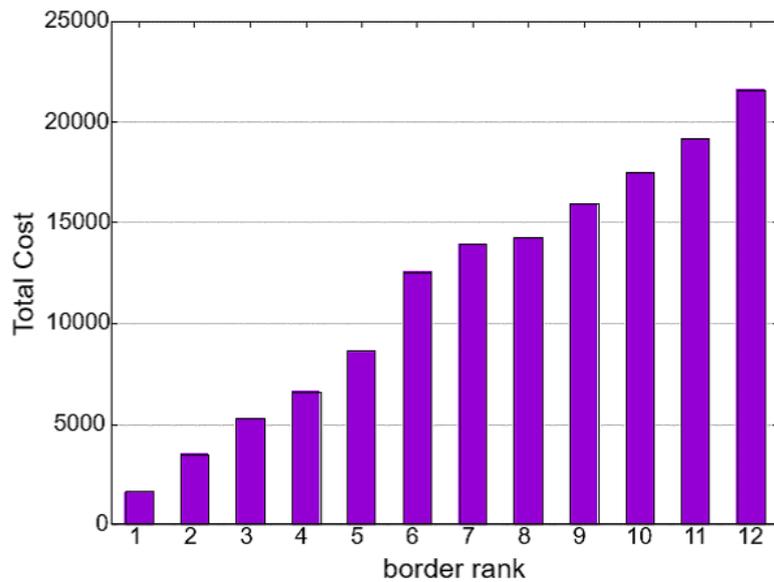


図 4.5: 上限移動コストが 2 のときの総移動コスト

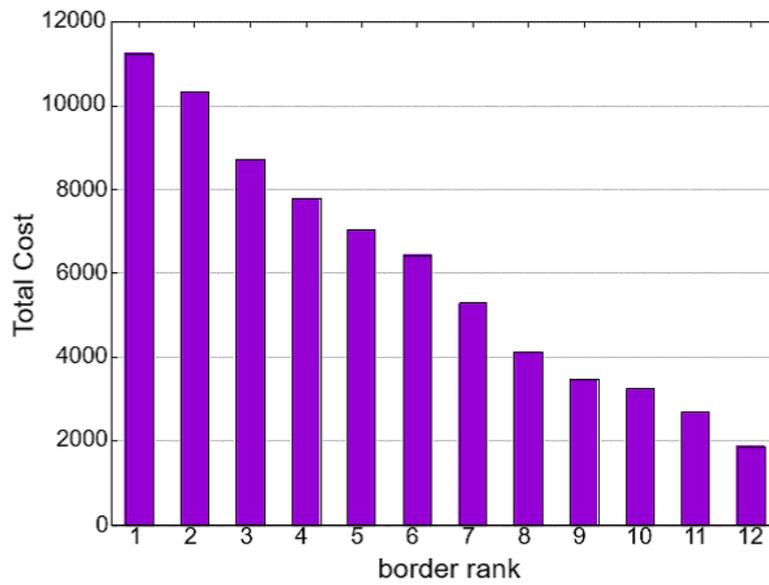


図 4.6: 上限移動コストが 1 のときの不動コンテンツ数

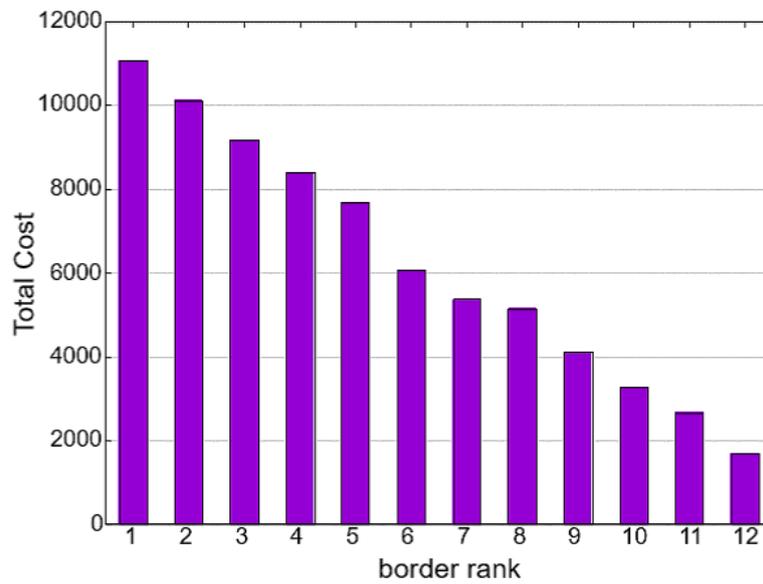


図 4.7: 上限移動コストが 2 のときの不動コンテンツ数

## 第5章 まとめ

この論文では、ICN の FIB メモリコストの増大と FIB エントリの検索時間や消費電力が増大する問題に着目し、コンテンツの移動コストを考慮しつつ、ICN ルータの FIB エントリが効率的に削減できる手法を提案した。コンテンツのオリジナルが配置されているノードから再配置するノードまでのホップ数を移動コストとした。移動コストに上限を設定し、その上限に基づいて、各 TLD 別で各ノードに配置可能なコンテンツ数を調査し、TLD ごとにコンテンツ配置ノードの優先度ランキングを作成した。そして、優先度ランキングの上位からコンテンツの配置を行った。再配置の対象にするランクに制限を設けて、平均 FIB サイズと総移動コストに与える影響を明らかにした。その結果から、全ノード間の最大移動コストに対する移動コストの上限の比率と、最大ランク数に対する配置制限ランクの比率を同じにすることで、平均 FIB サイズと総移動コストが最適な値を示すことを確認した。今後は最適な配置制限ランクが得られる、より詳細な原因の究明と、最大ノード次数が多いネットワークトポロジにて最適な配置制限ランクの確実性を示す研究を行う予定である。

## 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。また日常，有益な議論をして頂いた研究室の皆様にも感謝します。

## 参考文献

- [1] Y. Sasaki, et al., Designing Content Placement of CDN for Improving Aggregation Effect of ICN FIBs, ICOIN 2020
- [2] A. Hoque, et al., NLSR: Named-data Link State Routing Protocol, ACM ICN 2013.
- [3] A. Afanasyev, et al., SNAMP: Secure Namespace Mapping to Scale ICN Forwarding, GIS 2015.
- [4] A. Detti, M. Pomposini, N. Blefari-Melazzi, and S. Salsano, Supporting the Web with an information centric network that routes by name, Elsevier Computer Networks, Vol. 56, No. 17, pp. 3705-3722, Nov. 2012
- [5] W. So, A. Narayana, and D. Oran, Named data networking on a router: Fast and DoS-resistant forwarding with hash tables, ACM/IEEE ANCS 2013.
- [6] O. Ascigil, S. Rene, I. Psaras, and G. Pavlou, On-Demand Routing for Scalable Name-Based Forwarding, ACM ICN 2018.
- [7] R. Chiochetti, D. Rossi, and G. Caroglio, Exploit the Known or Explore the Unknown? Hamlet-Like Doubts in ICN, ACM ICN 2012.
- [8] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, A Survey on Content-Oriented Networking for Efficient Content De-livery, IEEE Commun. Mag., vol.49, no.3, pp.121-127, Mar. 2011.
- [9] K. Katsaros, et al., On the Inter-domain Scalability of Route-by-Name Information-Centric Network Architectures, IFIP Networking 2015
- [10] A. Rodrigues, P. Steenkiste, A. Aguiar, Analysis and Improvement of Name-based Packet Forwarding over Flat ID Network Architectures, ACM ICN 2018.
- [11] T. Schmidt, S. Wolke, N. Berg, and M. Wahlisch, Let 's Collect Names: How PANINI Limits FIB Tables in Name Based Routing, IFIP Networking 2016.
- [12] M. Gritter and D. R. Cheriton, An architecture for content routing support in the Internet, USENIX USITS 2001.
- [13] V. Jacobson, et al., Networking Named Content, ACM CoNEXT 2009.
- [14] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, A data-oriented (and beyond) network architecture, ACM SIGCOMM 2007.
- [15] G. Xylomenos, et al., A Survey of Information-Centric Networking Research, IEEE Communications Survey and Tutorials, Vol. 16, No. 2, pp.1024-1049, 2014.
- [16] L. Zhang, et al., Named Data Networking (NDN) Project, Technical Report NDN- 0001, Oct. 2010.
- [17] Y. Tang, K. Guo, Z. Liang, D. Liu, A Cooperative Replica Placement Strategy for MP2P-CDN, ISPA/IUCC 2017.
- [18] Q. Shi, X. Wang, W. Lin, Placement Strategy for Replicated Servers in CDN, CCC 2018.
- [19] Alexa webpage, <https://www.alexacom/siteinfo>
- [20] MaxMind, GeoIP Downloadable Databases, <https://dev.maxmind.com/geoip/legacy/downloadable>
- [21] Internet 2, <https://www.internet2.edu>