一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NS2019-133(2019-11)

ICN の FIB 集約のためのコンテンツ配置制御

佐々木　優†　　上山　憲昭†

† 福岡大学工学部電子情報工学科
〒 814–0180 福岡市城南区七隈 8–19–1
E-mail: †tl161252@cis.fukuoka-u.ac.jp, ††kamiyama@fukuoka-u.ac.jp

あらまし　デジタルコンテンツや IoT データを効率的に配信できる新しい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．しかし多くの場合，コンテンツの名称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．一方，インターネットでは CDN (content delivery network) が，ユーザの配信品質を向上しネットワーク内のトラヒック量を削減する技術として広く用いられているが，CDN のこれらの目的は ICN により達成される．そこで本稿では，CDN を ICN のオリジナル提供プラットフォームとして位置づけ，CDN のキャッシュサーバにコンテンツのオリジナルを，ICN ルータの FIB のサイズを効果的に削減するよう配置することを提案する．Web コンテンツの配置に関する測定データを用いた数値評価により，提案方法が FIB のサイズを平均で約 45%程度，削減できることを示す．

キーワード　ICN，FIB 集約，コンテンツ配置

Designing Content Placement for Improving Aggregation Effect of ICN FIBs

Yu SASAKI† and Noriaki KAMIYAMA†

† Department of Electronic and Information Technology，Fukuoka University
8–19–1，Nanakuma，Jounan，Fukuoka 814–0180
E-mail: †tl161252@cis.fukuoka-u.ac.jp, ††kamiyama@fukuoka-u.ac.jp

Abstract　Information-centric networking (ICN) has attracted wide attention as a new network architecture which can efficiently deliver digital content and Internet of Things (IoT) data. However, prefix aggregation of FIB (forwarding information base) of ICN routers is difficult because the organization names are independent of location. In this paper，we propose to use the CDN (content delivery network) as a method of allocating originals of content at routers to effectively reduce the size of FIBs of ICN routers. Using the measured data of web-content location，we evaluate the proposed algorithm of allocating web content at routers and show that the proposed method can reduce the size of FIBs by about 45% on average.

Key words　ICN，FIB aggregation，content placement

1. はじめに

YouTube などのユーザ生成コンテンツや，映画やドラマなどのコンテンツプロバイダによって生成されたリッチコンテンツの配信で生成されるトラヒックは，インターネット上のトラヒックの大部分を占めている．さらにスマートホーム，スマートビル，スマートシティなどで用いられる IoT (Internet of Thing) の実用化が進んでいるが，大量の IoT デバイスから膨大な量のセンシングデータをネットワークは転送する必要がある．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，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 ICN（information-centric networking）が注目されている [6]．ICN の概念を実現するために，TRIAD [8]，CCN (content-centric networking) [12]，DONA (data-oriented network architecture) [14]，NDN (named data networking) [21] などの様々なネットワークが提案されている [20].

本稿では ICN のアーキテクチャとして NDN を想定するが，NDN ではコンテンツを要求するパケットは Interest と呼ばれる．インターネットの IP ルータと同様に，NDN ルータは FIB（forwarding information base）を参照することで Interest を転送する．IP ルータでは転送先 IP アドレスの Prefix，つまりネットワークアドレスと出力ポートの組が FIB のエントリであるのに対し，NDN ルータではコンテンツ名の Prefix と出力ポートの組が FIB のエントリとなる．コンテンツ名は，Prefix とコンテンツ ID の連結であり，例えば，“fukuoka-u/sample.jpg”

の場合は “fukuoka-u” が Prefix，“sample.jpg” がコンテンツ ID となる．NDN ではコンテンツのオリジナルは転送元のホストに存在し，転送元を収容するルータはコンテンツ名の Prefix を隣接ルータに広告する [10].Prefix 広告を受信した NDN ルータは，ユーザである Subscriber から送信された Interest が，要求コンテンツのオリジナルを収容する 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が大幅に増加する．したがって，FIB のメモリコストを削減するには FIB エントリ集約が不可欠である．ネットワーク事業者は，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．一方，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があるが，組織の名前に地理的な地域性はない．したがって，NDN の Prefix 集約は IP ネットワークの Prefix 集約と比較して困難である．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 に集約した後であっても，約 10^9 のコンテンツ名の Prefix がある [7]. 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 [7] が，IP ネットワークの各 FIB には約 10^5 エントリしか必要ない．ハッシュ表 [19] に基づいたエントリ検索を行う場合，数百万の Prefix に対しても数百万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，多数のネットワーク上に配置したキャッシュサーバからコンテンツを配信する CDN（content delivery network）が，コンテンツ配信の品質を改善し，ネットワーク内のトラヒック量を削減するために広く使用されている [16]. NDN を用いた場合，配信要求ユーザの近くに存在するルータからコンテンツが配信される可能性が高いため，CDN の目的であるユーザ品質の向上とネットワークトラヒック量の削減は NDN によって満たされる．ところで CDN のキャッシュサーバも，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り [3]，CDN のキャッシュサーバは Publisher のホストになることもできる．そこで本稿では，CDN を NDN のコンテンツオリジナルの提供プラットフォームとして位置づけ，NDN ルータの FIB エントリの集約効果が向上するよう，コンテンツのコピーを CDN のキャッシュサーバに配置することを提案する．本稿の貢献は以下にまとめられる．

- 高人気の 8,000 の Web ページを閲覧し，これら Web ページを構成する Web オブジェクト(注1)の場所を測定し，各 Web オブジェクトを Internet2 のトポロジの 12 ノードのいずれかに割り当て，Web オブジェクトのコンテンツの配置場所に関するデータセットを作成する．
- NDN ルータの FIB エントリの集約アルゴリズムを提案し，測定された Web オブジェクトの場所に対して本アルゴリズムを適用した場合の FIB エントリの集約度を評価する．
- ネットワークノードに Web コンテンツを割り当てるアルゴリズムを提案し，提案方式により NDN ルータの FIB サイズを平均で約 45%削減できることを示す．

以下，2. 節で関連研究を簡単に述べ，3. 節では Web オブジェクトの位置測定手順を示す．そして 4. 節で NDN ルータの FIB エントリの集約アルゴリズムを提案し，測定された Web オブジェクトの位置を用いた数値評価結果を示す．そして 5. 節で FIB サイズを削減するためにネットワークノードに Web オブジェクトを配置するアルゴリズムを提案し，その数値評価結果を示す．最後に 6. 節で全体をまとめる．

2. 関 連 研 究

ICN ルータの FIB に必要なエントリ数を削減する方式を，(i) 部分キャッシュ，(ii) ルート集約，(iii) フラッディング，(iv) ブルームフィルタの 4 つのカテゴリに分類してまとめる．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 [1] [7]．Afanasyev らは DNS を使用し，エントリがルータの FIB に存在しない場合の名前解決法を提案している [1]．到着する Interest の Prefix の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 DNS を用いて，要求コンテンツと同じドメインを有しルータの FIB にエントリが存在する Prefix を得，その Prefix に対し Interest を送信する．また Detti らは，ルックアンドキャッシュアプローチを提案している．つまりルータは 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 [7]．本アプローチでは 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 [18]．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，ルータで必要な FIB のサイズが削減されるが，Interest の転送ホップ長は増加する．3 番目のアプローチ，すなわちフラッディングでは，FIB 検索を行わずにルータのすべての出力フェース(注2)に Interest がブロードキャストされる [3] [5].Ascigil らはルータで 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 [3]. Chiocchetti らは，人気のないコンテンツアイテムの Interest を転送し，FIB を使用せずに人気のあるコンテンツアイテムの Interest をブロードキャストすることを提案した [5]．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．

最後に 4 番目のアプローチ，つまりブルームフィルタでは，ルータが各出力フェースで提供されるブルームフィルタを使用し，到着する各 Interest を各出力フェースに転送するか否かを判断する [13] [17]．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 転送の決定が可能である．ただし Interest を誤った出力面に転送する可能性がり，冗長な Interest 送信によりネットワーク負荷が増加する．

3. Web オブジェクトの位置測定手順

adult，arts，business，computers，games，health，home，kids&teens，news，recreation，reference，regional，science，shopping，society，sports の 16 のカテゴリごとに，Web ページのアクセス数のランキングを公開している Alexa の Web

(注1)：Web オブジェクトは Web ページに埋め込まれたデータの単位であり，各 Web ページは平均で約 50〜100 個の Web オブジェクトで構成されている．

(注2)：NDN ではルータの出力ポートを出力フェースと呼ぶ．

ページから，各カテゴリに対して上位 500 の Web ページの URL を測定対象として選択した．Web ページにアクセスする場合，Google Chrome などの一部の Web ブラウザには各オブジェクトのダウンロード元のホスト URL，各オブジェクトの URL など，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 [4] [9]．HAR ファイルには，各オブジェクトのサイズや取得に要した遅延時間等，様々な情報が JavaScript Object Notation (JSON) 形式として出力される．2017 年 11 月に，生成した 8,000 の評価 URL リストの各 URL に対し，福岡大学の測定用クライアント端末から，GUI を使用しないヘッドレス Chrome を使用して，継続的かつ自動的にアクセスした．アクセスした 8,000 の Web ページのうち 7,604 の Web ページから HAR ファイルを正常に取得した(注3)．

表 1　Number of URLs in each of top 10 countries

Country	URL count	Country	URL count
United States	11，908	Czech Republic	237
Germany	477	France	228
Korea	322	Hong Kong	222
United Kingdom	261	Australia	202
Singapore	241	Ireland	128

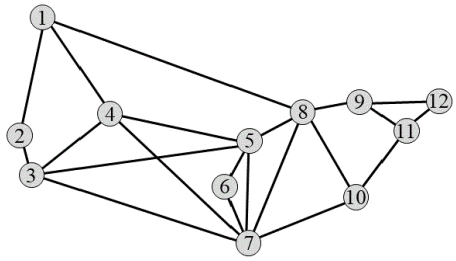


図 1　Layer 3 node topology of Internet2

表 2　Number of web objects allocated to each node

n	City	#Objects	n	City	#Objects
1	Seattle	1,267	7	Houston	265
2	Sunnyvale	1,997	8	Chicago	439
3	Los Angeles	623	9	Cleveland	237
4	Salt Lake City	602	10	Atlanta	322
5	Kansas City	3,050	11	Washington DC	1,611
6	Dallas	376	12	New York	1,119

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，HAR ファイルからこれらのオブジェクトの URL を抽出した．CDN はインターネットで広く使用されているため，多くの Web 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．しかしコンテンツのオリジナルの場所に関する情報を収集するため，キャッシュサーバから配信された URL を除外する．CDN を使用する場合，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” という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．したがって抽出された各 URL に対し，dig コマンドを使用して DNS レコードを取得し，DNS レコードに “CNAME” の文字列が含まれる URL を除いた 20,921 の URL を取得した．さらに，CDN に関連する 5 つのキーワード cdn，edge，akam，cloudfront，cloudflare を含む URL をさらに除外した．2,374 の URL にはこれらのキーワードのいずれかが含まれているため，最終的にオリジンサーバから Web オブジェクトが配信されていると予想される 18,547 の URL を取得した．

MaxMind [15] が提供する GeoIP API を使用して，18,547 個のオブジェクトのそれぞれが送信されたホストの国名，都市名，および座標を取得した．各オブジェクトが存在する国に基づいて 18,547 個のオブジェクトを分類したときの，上位 10 か国のそれぞれに分類された URL の数を表 1 に示す．18,547 個の Web オブジェクトのオリジナルが 71 か国から提供され，約 64%の Web オブジェクトがアメリカから提供されている．以下の評価では，アメリカに存在するオリジナル 11,908 個の座標と URL を使用する．

数値評価では，研究と教育を目的としたアメリカのバックボーンネットワークである Internet 2 のトポロジを使用した [11]．Layer 3 service を提供する 12 のノードを使用し，図 1 に選択した 12 のノードから構成される Internet 2 のトポロジを示す．O_m を，インターネット 2 トポロジの 12 個のノードの中で，Web オブジェクト m のオリジナルが配置されたノードと定義する．オブジェクト m を配信するホストの座標からのユークリッド距離が 12 のノードの中で最小であったノードを O_m に設定した．表 2 に，11，908 個のオブジェクトのうち，12 個の各ノードに割り当てられた Web オブジェクトの数をまとめる．多くの Web オブジェクトが西海岸や東海岸，そして北米の中央に位置するカンザスシティに割り当てられた．

4. NDN ルータの FIB エントリ集約

本節では LPM(longest prefix matching) を使用して NDN ルータの FIB エントリを集約するアルゴリズムを提案する．

4.1 LPM を使用した FIB 集約

IP ルータでは，FIB サイズを効率的に削減するために LPM が使用される．IP ルータの FIB エントリは，任意の長さの最下位ビットで任意の値，つまり 0 または 1 をとることができる．IP ルータに到着したパケットは，FIB エントリが到着パケットの宛先 IP アドレスと一致する最上位ビットの数が最大である出力ポートに転送される．LPM を NDN ルータに適応するには (i)URL の文字の単位，もしくは (ii)URL のコンポーネントの単位，の 2 つのアプローチが考えられる．ただしコンポーネントを，URL のピリオドで区切られた文字列と定義する．例えば “www.fukuoka-u.ac.jp” の URL は 4 つのコンポーネント，“www”，“fukuoka-u”，“ac”，“jp” で構成される．NDN の Prefix は，逆の順序でのコンポーネントの連結，つまり，“jp/ac/fukuoka-u/www ” となる．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) である．本稿では URL のコンポーネントを LPM の単位とする．

4.2 FIB エントリの集約アルゴリズム

最初に各ノード 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すべてのオブジェクトに対して FIB エントリを作成する．提案 FIB 集約アルゴリズムは各ノードで独立に実行できるため，以下では 1 つのノードに焦点を当て，ノードインデックス n を用いないですべての変数を記述する．FIB のエントリ x の Prefix を $P(x)$ で，$P(x)$ の最初の s 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

(注3)：一部のオブジェクトの取得時にタイムアウトエラーの発生，オブジェクトの一部の URL の IP アドレスを解決不可能等の理由により，396 の Web ページの HAR ファイルを取得でなかった．

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．Interests のルーティングアルゴリズムは本稿の範囲外であり，$F(x)$ が与えられていると仮定する．また，$\boldsymbol{M}$ を，未チェックの FIB エントリの集合と定義し，URL のコンポーネントの最大数を C とする．評価に用いる 11,908 個の Web オブジェクトでは C は 6 であった．以下に FIB エントリを順方向に集約するアルゴリズム（Algorithm 1）を記述する．

Algorithm 1 Aggregating FIB entries in forward direction

1: Initializes $s = 1$ and $\boldsymbol{M}$ to all FIB entries
2: **while** $s \leqq C - 1$ **do**
3: **while** $\boldsymbol{M} \neq \phi$ **do**
4: Randomly selects one entry x from $\boldsymbol{M}$
5: Calculates $\boldsymbol{Y}(x, s, o)$, set of FIB entry y satisfying $p(y, s) = p(x, s)$ with $F(y) = o$ among $y \in \boldsymbol{M}$, for each possible o
6: Aggregates all entries of $\boldsymbol{Y}(x, s, o^*)$ to FIB entry $\{p(x, s), o^*\}$ where number of entries of $\boldsymbol{Y}(x, s, o^*)$ is maximum among those of $\boldsymbol{Y}(x, s, o)$ of any output face o
7: Updates $\boldsymbol{M}$ to $\boldsymbol{M} \setminus \{y \in Y(x, s, o), \forall o\}$
8: **end while**
9: Increments s
10: **end while**

最初に TLD を対象に FIB エントリを集約し ($s = 1$)，次のステップでは TLD と SLD の組に対し FIB エントリを集約する ($s = 2$)．すべてのコンポーネントの位置 1，2，$\cdots$，$C-1$ での集約が各々，完了するまで本手順が繰り返される．また FIB エントリを逆方向，すなわち $s = C - 1$ から $s = 1$ の順に集約することも可能である．この場合，上記と同じアルゴリズムを使用して，ステップ 1 で $s = C - 1$ を設定し，ステップ 9 で s をデクリメントし，s が 1 以上である間はステップ 3〜9 を繰り返す．

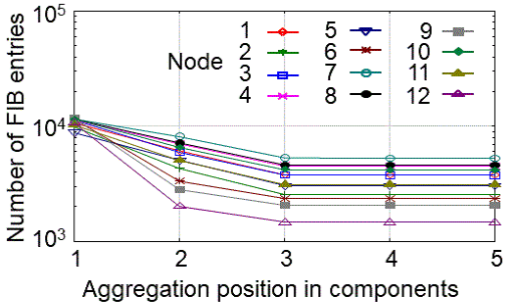
表 3 Number of FIB entries without aggregation, $E^I_{o,n}$, with aggregation in forward direction, $E^{A,f}_{o,n}$, and with aggregation in reverse direction, $E^{A,r}_{o,n}$, and $\Delta E^A_{o,n} \equiv E^{A,f}_{o,n} - E^{A,r}_{o,n}$ based on measured location of web objects

Node	$E^I_{o,n}$	$E^{A,f}_{o,n}$	$E^{A,r}_{o,n}$	$\Delta E^A_{o,n}$
1	10,641	3,792	3,646	146
2	9,911	2,554	2,533	21
3	11,285	3,746	3,583	163
4	11,306	4,528	4,550	-22
5	8,858	3,043	3,011	32
6	11,532	2,360	2,393	-33
7	11,643	5,289	5,207	82
8	11,469	4,567	4,509	58
9	11,671	2,061	2,083	-22
10	11,586	4,196	4,208	-12
11	10,297	3,130	3,139	-9
12	10,789	1,462	1,481	-19

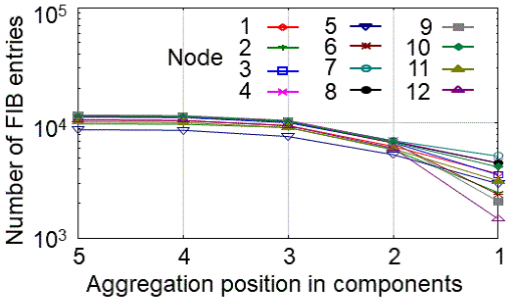
4.3 数値評価

前節で提案した集約アルゴリズムを 11,908 個の Web オブジェクトの実測によるオリジナル位置に対し適用した．任意のノードペア間に対し最小ホップ経路の使用を想定した．$E^{I}_{o,\,n}$，$E^{A,\,f}_{o,\,n}$，および $E^{A,\,r}_{o,\,n}$ を各々，集約なし，順方向のエントリ集約後，逆方向のエントリ集約後のノード n の FIB エントリ数とする．表 3 に，$E^{I}_{o,\,n}$，$E^{A,\,f}_{o,\,n}$，$E^{A,\,r}_{o,\,n}$ を示す．各ノード n に対し，$\Delta E^{A}_{o,\,n}$ を $E^{A,\,f}_{o,\,n}$ と $E^{A,\,r}_{o,\,n}$ の差，$E^{A,\,f}_{o,\,n} - E^{A,\,r}_{o,\,n}$ と定義する．測定により得られた Web オブジェクトの位置においても，提案集約アルゴリズムを用いることで，各ルータの FIB サイズを約 20%から 50%に減らすことが可能である．両方向の集約効果の差は小さいが，FIB エントリを集約するのに必要な計算時間は逆方向よりも順方向の方がはるかに短いため，順方向の集約が逆方向よりも優れている．

図 2(a) は順方向に FIB エントリを集約したときの URL コンポーネントの集約各時点での，12 個の各ノードの FIB エントリ数を示す．また図 2(b) に FIB エントリを逆方向に集約した場合の結果を同様に示す．最初と 2 番目のコンポーネント，すなわち TLD と SLD で FIB エントリを集約することで，FIB エントリのほぼ全ての集約効果が得られることが確認できる．



(a) In forward direction



(b) In reverse direction

図 2 Number of FIB entries at each node based on measured location of object originals

5. FIB サイズの効率的削減を目的とした CDN のコンテンツ配置設計

5.1 CDN のコンテンツ配置の設計方針

前節で述べたように，FIB エントリのほとんどすべての集約効果は TLD と SLD で得られる．そのため TLD と SLD のみでの FIB エントリの集約を考慮した，CDN キャッシュサーバのコンテンツ配置設計を考える．図 3 に，評価に用いた 11,908 個の Web オブジェクトにおいて，同一の TLD と同一の SLD を用いた Web オブジェクト数の相補累積分布示す．11,908 個の Web オブジェクトの異なる TLD および SLD の数は，各々 112 と 6,494 である．多くの Web オブジェクトの URL で使用される TLD と SLD は非常に少ないのに対し，ほとんどすべての TLD と SLD は少数の Web オブジェクトの URL でのみ使用される．表 4，5 に各々，使用 URL 数の上位 10 個の TLD および上位 10 個の SLD を有する Web オブジェクトの URL の数 N_d をまとめる．各 TLD を使用した URL 数の偏りは，各 SLD を使用した場合と比較して著しく大きく，11,908 個の Web オ

ブジェクトの約 70%の URL の TLD は “com” であった．

次に各 Web オブジェクトを 12 のノードのいずれかに配置することを検討する．オリジナルが同じノードに存在する Prefix に対しては，各ノードの FIB において同一の出力フェースが設定される．したがって”com” 等の順位が高い TLD を有する Web オブジェクトを同一ノードに配置することにより，FIB エントリ数を効率的に減らすことができる．極端なケースでは，すべての Web オブジェクトが 1 つのノードに配置され，すべての FIB エントリが 1 つのエントリに集約された場合である．ただしすべての Web オブジェクトとそのリンクが接続されたノードにトラヒック負荷が集中するため，トラヒック量を緩和するには 12 個すべてのノードに可能な限り均等に Web オブジェクトを配置する必要がある．したがって，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 B 以下に制限する．B はノード数 N で除した Web オブジェクト数 W 以上でなければならない．集約後の各ノードの FIB サイズは，ネットワークトポロジ上の Web オブジェクトのオリジナルの位置に依存する．ただし本稿では，ネットワークトポロジ上の各ノードの位置は考慮せず，ノードの位置を考慮した FIB 集約法は今後の課題とする．

図 3　Complementary cumulative distribution of web object count with identical TLD and identical SLD

表 4　Number of URLs with each of top 10 TLDs

Rank	TLD	N_d	Rank	TLD	N_d
1	com	8,241	6	io	122
2	net	1,292	7	uk	115
3	org	953	8	au	75
4	edu	441	9	co	64
5	gov	181	10	ca	54

表 5　Number of URLs with each of top 10 SLDs

Rank	SLD	N_d	Rank	SLD	N_d
1	com/amgdgt	397	6	au/com	59
2	net/omtrdc	395	7	net/2o7	51
3	net/openx	193	8	com/gstatic	44
4	uk/co	100	9	com/netdna-ssl	36
5	com/mktoresp	73	10	net/fastly	36

5.2　CDN のコンテンツ配置アルゴリズム

5.1 節で述べた方針に基づき，FIB サイズの低減を目的としたコンテンツ配置アルゴリズム(注4)を提案する．ネットワークのすべてのノードに CDN のキャッシュサーバが提供されている状態を想定する．使用 URL 数の降順で TLD をソートし，$D_1(x)$ を x 番目にランクされた TLD と定義する．同様に各 TLD に対し，URL 数の降順で SLD の組をソートし，$D_2(s,\ y)$ を TLD が s で y 番目にランクされた SLD と定義する．M_1 を TLD の異なり数と，$M_2(s)$ を TLD s を有する SLD の異なり数と定義する．さらに，$\boldsymbol{U}_1(x)$ は TDL として $D_1(x)$ を持つ URL の数を示し，$\boldsymbol{U}_2(s,x)$ は TLD として s を，SLD として $D_2(s,x)$ を有する URL の数を示す．さらに $m_1(x)$ と $m_2(s,x)$ を，$\boldsymbol{U}_1(x)$ と $\boldsymbol{U}_2(x)$ に含まれる URL の数として定義する．A_n はノード n に配置できる URL の数を示し，n^* は A_n の最大値を持つノードを示す．以下に，FIB サイズの低減を目的としたコンテンツの配置アルゴリズム (Algorithm 2) を示す．

Algorithm 2 Placing content at nodes for reducing FIB size

```
Initializes x = 1 and A_n = B for all n
while x ≦ M_1 do
  Finds n*
  if m_1(x) ≦ A_{n*} then
    Places U_1(x) at node n* and subtracts m_1(x) from A_{n*}
  else
    Initializes y = 1
    while y ≦ M_2(D_1(x)) do
      while m_2(D_1(x), y) > A_{n*} do
        Places U_2(D_1(x), y) at node n* and subtracts
        m_2(D_1(x), y) from A_{n*}
      end while
      Increments y and finds n*
    end while
  end if
  Increments x
end while
```

$m_1(x)$ の降順に，$m_1(x) \leqq A_{n^*}$ の場合にはノード n^* に $\boldsymbol{U}_1(x)$ を割り当て，$m_1(x) > A_{n^*}$ の場合には $\boldsymbol{U}_1(x)$ を $y = 1, 2, \cdots, M_2(D_1(x))$ として $\boldsymbol{U}_2(D_1(x), y)$ に分割し，$\boldsymbol{U}_2(D_1(x), y)$ 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 の数であるとき，$u_n \leqq B$ を満たしながら，各ノードの FIB エントリの効果的な集約が期待できる．

5.3　数 値 評 価

1 つのノードに配置されたオブジェクト数の上限である B を $W/N * 1.1 = 1,091$ に設定した．ただし W は Web オブジェクトの数 11,908 で，N はノードの数 12 である．表 6 に，各ノードに割り当てられた TLD，SLD，および URL の数を示す．表 4 で示したように，約 70%の URL の TLD は “com” であり，これら 8,241 個の URL は SLD 別にノード 1, 2, ⋯, 8 に割り当てられた．ノード 11 では TLD が “org” であるすべての URL が割り当てられた．これらの結果から，提案アルゴリズムは，上限が $u_n \leqq B$ である限り同じノードに，同一の TLD または SLD を持つ URL を割り当てることが確認できる．

次に，4.2 節で述べた Algorithm 1 を Algorithm 2 で得られた Web オブジェクト配置に適用した．表 7 に，提案方式で設計した Web オブジェクト配置における，集約なしの FIB エントリの数 $E^I_{d,n}$，順方向に集約された FIB エントリの数 $E^{A,f}_{d,n}$，逆方向に集約された FIB エントリの数 $E^{A,r}_{d,n}$ を示す．さらに表には，$\Delta E^A_{d,n} \equiv E^{A,f}_{d,n} - E^{A,r}_{d,n}$ で定義される $E^{A,f}_{d,n}$ の縮小率と，$E^{A,r}_{d,n}$ の縮小率 $R_{n,r}$ をまとめる．$R_{n,f}$ と $R_{n,r}$ は，Algorithm 2 で得られた集約後のノード n での FIB エントリ数の削減率の，Web オブジェクトの実測配置に対して Algorithm 1 得られ

(注4)：本稿では content という用語を web object と同じ意味で用いる．

た削減率に対する比較で定義し，$R_{n,f} = (E^{A,f}_{o,n} - E^{A,f}_{d,n})/E^{A,f}_{o,n}$，$R_{n,r} = (E^{A,r}_{o,n} - E^{A,r}_{d,n})/E^{A,r}_{o,n}$ で与えられる．

提案方式で Web オブジェクトを配置設計した場合，$\Delta E^{A}_{d,n}$ はゼロ以上であり，FIB エントリを逆方向に集約することは順方向よりも効果的であった．表 2 に示したように，実測による得られた Web オブジェクト配置において，ノード 5 に配置された Web オブジェクト数は他のノードのそれよりもはるかに多い．一方，ノード 5 に割り当てられたオブジェクトの数は，提案方式でオブジェクトを配置した結果，他のノードのオブジェクトの数とほぼ等しくなる．そのため実測によるオブジェクト配置におけるノード 5 の FIB エントリの初期数 $E^{I}_{o,5}$ は，提案方式での割り当て後の数 $E^{I}_{d,5}$ よりもはるかに小さい．したがって提案 Algorithm 2 によって設計されたオブジェクトの場所での集約後のノード 5 での FIB エントリ数は，実測におけるオブジェクト配置でのエントリ数よりも多くなる．しかし他のすべてのノードでは，提案方式でオブジェクトを配置することで，FIB エントリ数を約 20%から約 90%削減できた．エントリ集約後の各ノードでの平均 FIB サイズは，実測による Web オブジェクト配置では 3,394.0 であったが，提案方式での設計配置では 1,848.4 であった．提案方式で Web オブジェクトの配置を設計することで，FIB サイズを平均で約 45%削減できた．

表 6　Number of TLDs, SLDs, and URLs assigned to each node

Node	#TLD	#SLD	#URL	Node	#TLD	#SLD	#URL
1	0	65	1,090	7	0	1,091	1,091
2	0	315	1,090	8	35	608	744
3	0	486	1,090	9	0	145	1,091
4	0	545	1,090	10	38	198	744
5	0	630	1,091	11	1	0	953
6	0	1,091	1,091	12	36	0	743

表 7　Number of FIB entries without aggregation, $E^{I}_{d,n}$, with aggregation in forward direction, $E^{A,f}_{d,n}$, with aggregation in reverse direction, $E^{A,r}_{d,n}$, $\Delta E^{A}_{d,n} \equiv E^{A,f}_{d,n} - E^{A,r}_{d,n}$, reduction ratio of $E^{A,f}_{d,n}$, $R_{n,f}$, and reduction ratio of $E^{A,r}_{d,n}$, $R_{n,r}$, based on location of web objects designed by proposed method

Node	$E^{I}_{d,n}$	$E^{A,f}_{d,n}$	$E^{A,r}_{d,n}$	$\Delta E^{A}_{d,n}$	$R_{n,f}$	$R_{n,r}$
1	10,818	1,521	1,521	0	0.599	0.583
2	10,818	1,330	1,330	0	0.479	0.475
3	10,818	2,326	2,273	53	0.379	0.366
4	10,818	3,608	2,822	786	0.203	0.38
5	10,817	3,710	3,367	343	-0.219	-0.118
6	10,817	1,401	1,348	53	0.406	0.437
7	10,817	3,249	2,906	343	0.386	0.442
8	11,164	2,291	2,238	53	0.498	0.504
9	10,817	112	112	0	0.946	0.946
10	11,164	1,062	1,062	0	0.747	0.748
11	10,955	1,297	1,244	53	0.586	0.604
12	11,165	274	221	53	0.813	0.851

6. ま と め

本稿ではまず，URL のコンポーネント単位で順方向および逆方向に FIB エントリを集約するアルゴリズムを提案した．Internet 2 のトポロジと Web オブジェクトの実測の配置を用いた数値評価により，TLD と SLD で集約することにより，ほぼすべての FIB エントリ数の集約効果が得られることを確認した．そこで FIB エントリの集約効果を効果的に改善することを目的に，同じ TLD または SLD を URL に有する Web オブジェクトを同じノードに割り当てる，Web オブジェクトの CDN キャッシュサーバへの割り当て方式を提案した．数値評価により，提案方式を用いることで，FIB サイズを平均で約 45%程度，削減できることを示した．

謝辞　本研究成果の一部は，KDDI 財団研究助成寄付金 190051 の助成を受けたものである．ここに記して謝意を表す．

文　献

[1] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang, SNAMP: Secure Namespace Mapping to Scale NDN Forwarding, IEEE Global Internet Symposium 2015.
[2] Alexa webpage, https://www.alexa.com/siteinfo
[3] O. Ascigil, S. Rene, I. Psaras, and G. Pavlou, On-Demand Routing for Scalable Name-Based Forwarding, ACM ICN 2018.
[4] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, Understanding Website Complexity: Measurements, Metrics, and Implications, ACM IMC 2011.
[5] R. Chiocchetti, D. Rossi, and G. Carofiglio, Exploit the Known or Explore the Unknown? Hamlet-Like Doubts in ICN, ACM ICN 2012.
[6] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, A Survey on Content-Oriented Networking for Efficient Content Delivery, IEEE Commun. Mag., vol.49, no.3, pp.121-127, Mar. 2011.
[7] A. Detti, M. Pomposinim 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.
[8] M. Gritter and D. R. Cheriton, An architecture for content routing support in the Internet, USENIX USITS 2001.
[9] Software is hard, http://www.softwareishard.com/blog/harviewer/
[10] A. Hoque, et al., NLSR: Named-data Link State Routing Protocol, ACM ICN 2013.
[11] Internet 2, https://www.internet2.edu
[12] V. Jacobson, et al., Networking Named Content, ACM CoNEXT 2009.
[13] K. V. Katsaros, W. K. Chai, and G. Pavlou, Bloom Filter based Inter-domain Name Resolution: A Feasibility Study, ACM ICN 2015.
[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] MaxMind, GeoIP Downloadable Databases, https://dev.maxmind.com/geoip/legacy/downloadable/.
[16] E. Nygren, R. K. Sitaraman, and J. Sun, The Akamai Network: A Platform for High-Performance Internet Applications, ACM SIGOPS 2010.
[17] A. Rodrigues, P. Steenkiste, A. Aguiar, Analysis and Improvement of Name-based Packet Forwarding over Flat ID Network Architectures, ACM ICN 2018.
[18] 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.
[19] 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.
[20] G. Xylomenos, et al., A Survey of Information-Centric Networking Research, IEEE Communications Survey and Tutorials, Vol. 16, No. 2, pp.1024-1049, 2014.
[21] L. Zhang, et al., Named Data Networking (NDN) Project, Technical Report NDN-0001, Oct. 2010.