

NDNルータの転送テーブル 集約効果向上のための コンテンツ配置制御

福岡大学
上山憲昭

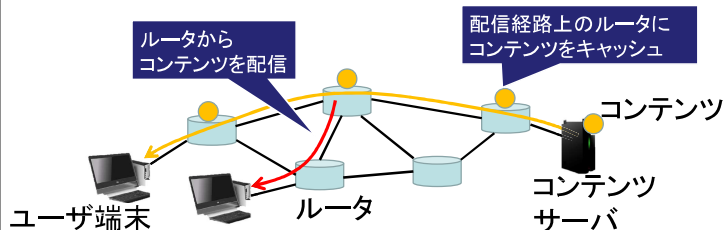
1

情報指向ネットワーク

■ 情報指向ネットワーク(NDN: named data networking)がコンテンツやIoTデータを効率的に配信するネットワークとして注目

■ NDNの特徴

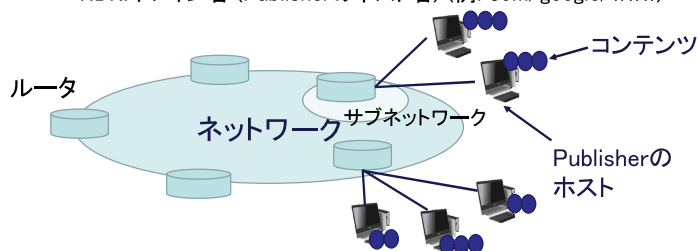
- IPアドレスを用いずコンテンツ名で配信要求をコンテンツのオリジナルに向け転送
- ルータでコンテンツをキャッシュし配信



2

NDNのコンテンツ名

- コンテンツ名 = Prefix + コンテンツ識別子
- Prefix: ルータがパケットの転送に用いる識別子
 - IPネットワーク: 目的ホストのサブネットワークアドレス
 - NDN: ドメイン名 (Publisherのホスト名) (例: com/google/www)



ドメイン数 >> サブネットワーク数

3

NDNの課題

- ドメイン数はサブネットワーク数と比較して膨大
- ドメイン名には地域性がないため、ルータの転送テーブル (FIB: forwarding information base)のエントリの集約が困難
- 例: Webコンテンツのみを考慮した場合
IP FIB: 約 10^5 のエントリ vs NDN FIB: 約 10^9 のエントリが必要*

IPルータのFIB		NDNルータのFIB		NH: next hop
Prefix	NH	Prefix	NH	
223.1.1.1	0	abc	0	
223.2.1.1	1	xy	1	
224.1.1.3	2	xyz	3	

- FIBの必要メモリ量と検索時間が大幅に増大

*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

4

本研究の着眼点

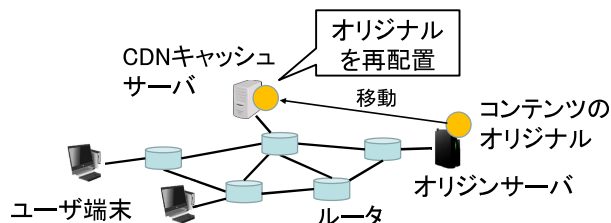
- 現在のインターネットは、コンテンツを効率的に配信する技術としてCDN(content delivery network)を広く利用
- CDNの本来の目的である、ネットワーク内トラヒック量と配信遅延時間の低減は、NDNによって達成
⇒ **CDNをNDNのオリジナル提供プラットフォームと位置付け***
- NDNのFIBエントリの集約可能性は、コンテンツのオリジナルの位置に依存 ⇒ **CDNを用いてコンテンツのオリジナルを再配置***

*佐々木優, 上山憲昭, "ICNのFIB集約のためのコンテンツ配置制御", 電子情報通信学会ネットワークシステム(NS)研究会, NS2019-133, 神戸, 2019年11月

5

本研究の目的

CDNをNDNのオリジナル提供プラットフォームと位置付け、NDNのFIBエントリ集約効果が効果的に向上するよう、CDNのキャッシュサーバ上のコンテンツ配置を設計



⇒ コンテンツのオリジナルの位置を、FIBサイズが効果的に集約できるよう変更

6

IPルータのFIB集約とLPM

- IPルータでは目的サブネットワークアドレスがPrefix
- ルータは各到着パケットに対しFIBを参照し、記載Prefixに対応するNH (next hop)にパケットを転送
- LPM (longest prefix matching)
ルータはPrefixの上位ビットが一致する長さが最大のFIBエントリを参照

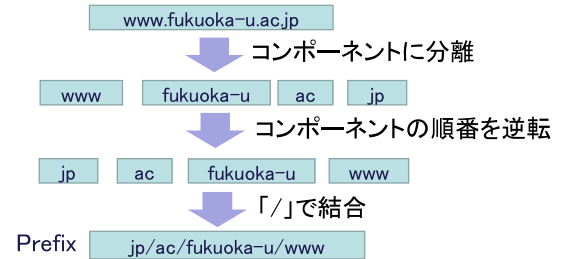
例: 223.1.1.1 ⇒ NH = 1
223.1.1.3 ⇒ NH = 0

Prefix	NH
223.1.1.*	0
223.1.1.1	1
224.1.1.3	2
224.1.2.1	3

7

NDNのPrefixの構成: コンポーネント

- コンポーネント: コンテンツの名前(URL)の「.」で区切られた各パート
- NDNのPrefix:
 - URLのドメイン部分をコンポーネントで分離
 - コンポーネントを逆順に並び替え, 「/」で結合
- 例: 「www.fukuoka-u.ac.jp」の場合



8

NDNのFIB集約方法の考え方

- Prefixをキーとしたハッシュ値に基づくFIBエントリ検索を想定
- 先頭から任意の個数のコンポーネントが完全に一致するエントリで、同一NHを有するものを1つに集約可能
- 下図: 上2つのエントリは、第1・第2コンポーネントが一致し、かつNH (next hop)が一致しているため集約可能

Prefix	NH		Prefix	NH
com/abc/xy	1	→	com/abc/*	1
com/abc/xyz	1	→	com/abc/yz	2
com/abc/yz	2	→		

集約前

集約後

9

NDNのFIB集約の範囲

- 先頭(第一)から任意の個数のコンポーネントの範囲とNHが一致するエントリを集約可能

先頭から4コンポーネントで集約

先頭から3コンポーネントで集約

com/abc/def/ghi/xyz
com/abc/def/ghi/tuv
com/abc/def/456/123

上の2エントリが集約可能

com/abc/def/ghi/xyz
com/abc/def/ghi/tuv
com/abc/def/456/123

全3エントリが集約可能

様々な集約範囲で集約可能

10

NDNのFIB集約の反復

- コンポーネントの様々な範囲で集約を反復

Prefix	NH		Prefix	NH
com/abc/xy	1	→	com/abc/xy	1
com/abc/xyz	1	→	com/abc/xyz	1
com/abc/yz	2	→	com/abc/*	2
com/abc/23	2	→		
com/abc/11	2	→		
com/abc/pq	3	→	com/abc/pq	3

- 着目範囲のコンポーネント列が一致するエントリ集合をNHごとに作成
- 該当エントリ数が最大の集合のみを1エントリに集約

FIBエントリの集約順番 { 短順: 集約範囲が短→長の順で反復
長順: 集約範囲が長→短の順で反復

11

FIB集約アルゴリズム

- 入力パラメタ:

- P(x): FIBのエントリxのPrefix
- p(x, s): P(x)の上位s個のコンポーネント
- F(x): FIBのエントリxのNH
- M: FIBの未検査エントリ集合

- アルゴリズム(短順):

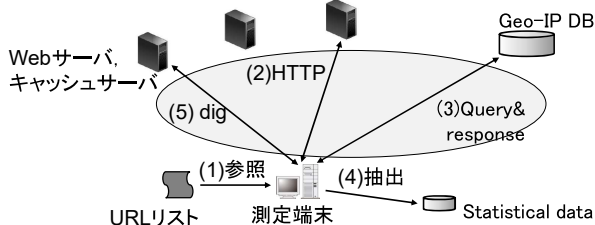
1. s = 1, M = {全FIBエントリ}に初期化
2. x ∈ Mのエントリxを1つ選択
3. y ∈ M, p(y, s) = p(x, s)のエントリ y でNHが o の集合 Y(x, s, o)を全ての o に対し計算
4. Y(x, s, o)の構成エントリ数が最大の o = o* に対し, Y(x, s, o*)のエントリを全てエントリ{p(x, s, o*), o*} に集約
5. Mを M \ {y ∈ Y(x, s, o) for all o} に更新
6. Mが空集合となるまで2~5を反復
7. sをインクリメントし, sが最大コンポーネント数C未満のときは2へ

12

Webオブジェクトの測定方法

- 数値評価には、WebオブジェクトのURLや位置を使用

1. アクセスURLリストを作成
2. 福大内の測定用端末から、URLリストに従いWebページにアクセスし、発生した通信の各種情報を含むHAR (HTTP Archive)ファイルを取得
3. HARファイル中の各オブジェクトのURL情報から、MaxMindのGeo IP-DBを参照し、各配信サーバの位置座標や都市名を取得
4. digを用いて各配信サーバのドメイン名を取得



13

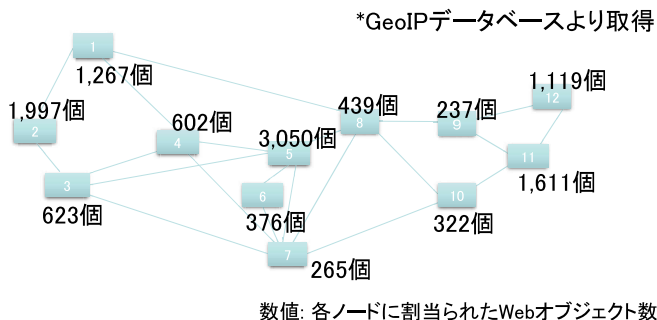
評価に用いたWebオブジェクトデータ

- AlexaのWebサイトのランキングで人気の8,000のWebページからWebオブジェクト679,380個を取得
- CDN配信と考えられるものを除外
 - DNSレコードにCNAMEを含むもの
 - URLのドメイン名にcdnやakamai等のCDN事業者に関連するキーワードを含むもの
- アメリカからの配信に限定 ⇒ 11,908個のWebオブジェクトを評価に使用

14

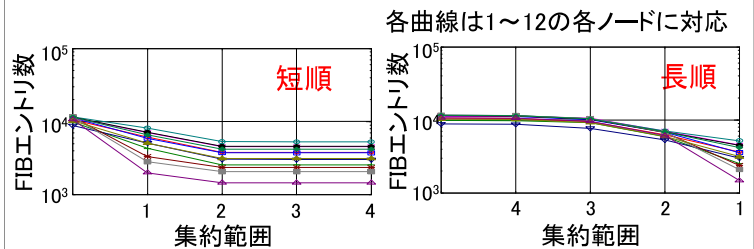
実測で得られたオブジェクト配置

- 評価に用いたネットワークポロジ: Internet2
- 11,908個の各Webオブジェクトを、その配信サーバの位置座標*に最も近いノードに割り当



15

実測オリジナル配置のFIBエントリ集約効果

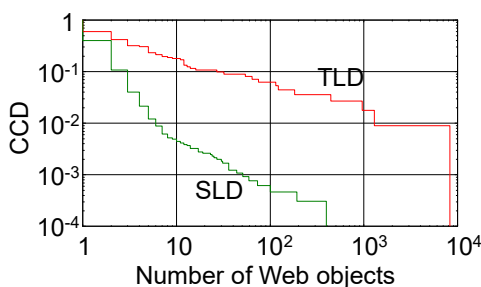


- 上位1(TLD: top level domain) or 2 (SLD: second level domain) 個のコンポーネントが同一のPrefix集約によって、ほとんどのFIBエントリ集約効果を達成 ⇒ TLD, SLDの集約を考慮すれば十分
- 実際のオリジナル配置においても、LPMにより1/2～1/5程度にFIBを集約可能
- 短順と長順の集約効果の差異は小さいが、計算に要する時間は短順の方が小

16

同一のTLD/SLDを有するWebオブジェクト数

- 11,908個のオブジェクトを、同一のTLDやSLDを有するもので分類 ⇒ 異なり数は、TLDが112個で、SLDが6,494個
- 各TLD or SLDに対し、分類されたURL数の累積補分布
 - 大多数のドメインは少数のURLのみ該当
 - ごく少数のドメインは極めて多数のURLが該当



17

TLD・SLDの上位リスト

該当OBJ数上位のTLD

com	8241
net	1292
org	953
edu	441
gov	181
io	122
uk	115
au	75
co	64
ca	54
tv	32
jp	27
info	16
nz	14
us	13
za	12
me	12
ai	12
ie	12
in	11

該当OBJ数上位のSLD

com/amgdgt	397
net/omtrdc	395
net/openx	193
uk/co	100
com/mktoresp	73
au/com	59
net/2o7	51
com/gstatic	44
com/netdna-ssl	36
net/fastly	36
com/amazonaws	36
com/micpn	31
com/parsely	30
net/doubleclick	27
gov/nih	25
com/Zendesk	24
net/blueconic	23
com/google	19
com/appspot	17
net/perimeterx	17
com/fragrancenet	17

- 11,908個のオブジェクトをTLDとSLDで分類し、分類数の上位をリストアップ
- TLDの偏りは大きく、最上位のcomを有するURLは全体の70%
- 上位のTLD/SLDを有するコンテンツを同一のノードに配置することでFIB集約効果の向上が期待



18

コンテンツ配置の設計方針

- FIBのエントリ集約効果を向上
⇒ 第1・2コンポーネントが同一のドメインに対する各ノードのNHを同一にしたい ⇒ **TLDやSLDが同じドメインのオブジェクトをできるだけ同一ノードに配置**
⇒ **TLDやSLDを各ノードに割当**
- 負荷の集中を避けるため、全ノードにできるだけ均等にオブジェクトを配置 ⇒ **各ノードに上限値以下の数のオブジェクトを配置**
- ノードの位置は考慮しない

19

コンテンツ配置アルゴリズム

- 方針: 各TLDやSLDをノードに割当
各ノードに配置するオブジェクト数を上限値B以下に抑制
- 入力パラメタ:
 - $D_1(x)$: TLDのリスト(該当オブジェクト数の降順でソート)
 - $D_2(s, x)$: TLDがsのSLDのリスト(該当オブジェクト数の降順でソート)
 - $n_1(x), n_2(s, x)$: $D_1(x), D_2(s, x)$ を有するオブジェクト数
 - $U_1(x), U_2(s, x)$: $D_1(x), D_2(s, x)$ を有するオブジェクトの集合
- アルゴリズム:
 1. 各ノードnの割当可能オブジェクト数を $A_n = B$ に、割当TLDを $x = 1$ に初期化
 2. A_n が最大のノード n^* を計算
 3. If $(n_1(x) \leq A_{n^*}) \Rightarrow U_1(x)$ をノード n^* に割当て、 $A_{n^*} \leftarrow n_1(x)$
 4. If $(n_1(x) > A_{n^*}) \Rightarrow$
 - I. $y = 1$ に初期化
 - II. while $(n_2(D_1(x), y) \leq A_{n^*}) \Rightarrow U_2(D_1(x), y)$ をノード n^* に割当て、 $A_{n^*} \leftarrow n_2(D_1(x), y)$
 - III. y をインクリメントし、 A_n が最大のノード n^* を計算
 - IV. TLDが $D_1(x)$ の全てのSLDを有するURLの割当が完了するまでII, IIIを反復
 5. x をインクリメントし、未割当TLDが存在すれば2へ

20

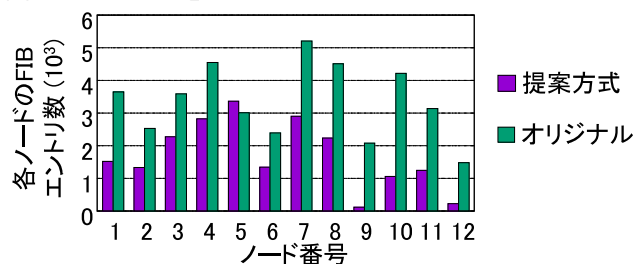
TLD/SLDのノードへの割当結果

- 各ノードの割当オブジェクト数の上限を
 $B = W/N \times 1.1 = 1,091$ に設定 (W: Webオブジェクト数, N: ノード数)
 - 各ノードの割当TLD/SLD数
- | Node | 割当TLD数 | 割当SLD数 | 該当オブジェクト数 | Node | 割当TLD数 | 割当SLD数 | 該当オブジェクト数 |
|------|--------|--------|-----------|------|--------|--------|-----------|
| 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 |
- 約70%のオブジェクトのTLDは「com」で、これら8,241個のオブジェクトはSLD別にノード1～8に割当
 - ノード11にTLDが「org」の全オブジェクトが割当

21

各ノードのエントリ集約効果

- 測定で得られたオブジェクト配置(オリジナル)状態と、提案方式(長順)でオブジェクトを配置した状態の各々に対し、FIBエントリ集約アルゴリズムを適用



- 提案方式を用いることでFIBエントリ数を3割～9割程度、削減
- オリジナル配置において、ノード5には多数のオブジェクトが配置されているが、提案方式により割当数が大幅に減少するため、提案方式によりFIBサイズは微増

22

提案オブジェクト配置法の効果

- 測定で得られたオブジェクト配置(オリジナル)状態と、提案方式でオブジェクトを配置した状態の各々に対し、FIBエントリ集約アルゴリズムを適用したときの、全ノードの平均FIBエントリ数を比較

集約前	オリジナル配置		提案方式による再配置	
	短順集約	長順集約	短順集約	長順集約
10,915.7	3,394.0	3,361.9	1,848.4	1,703.7

- 提案方式を用いることでFIBエントリ数を約45%削減

23

まとめ

- コンテンツの名前でパケットを転送するNDNがコンテンツやIoTデータを効率的に転送できるネットワークとして注目
- NDNではルータのFIBのサイズ低減が課題
- 本研究: CDNをNDNのコンテンツ提供プラットフォームと位置付け、FIBエントリ集約効果が効果的に向上するようコンテンツを配置設計
- 提案方式の効果: NDNルータのFIBサイズを約45%低減

24