

Osaka University

## OpenFlow を用いた CCN アーキテクチャ およびルータの実装

†大岡 睦, †阿多 信吾, †小出 俊夫, †下西 英之, †村田 正幸

†大阪大学 大学院情報科学研究科  
†大阪市立大学 大学院工学研究科  
†日本電気株式会社 クラウドシステム研究所

Osaka University

## 発表内容

- 研究背景
  - CCN の実現課題と OpenFlow
- 提案手法
  - OpenFlow を用いた CCN
- 実装方法
- 今後の課題

2013/5/17 5月 IN 研究会

Osaka University

## 研究背景

インターネットの設計思想と  
現在のネットワーク利用形態の不一致

- インターネットの設計思想
  - エンドホスト間の通信が主体
  - 通信がエンドホストの位置に依存
- 現在のネットワーク利用形態
  - コンテンツの取得・配布が主体
  - コンテンツを保持するエンドホストの位置には無関心

現在のコンテンツ中心の利用形態に沿った  
コンテンツセントリックネットワーク (CCN) へ

2013/5/17 5月 IN 研究会

Osaka University

## コンテンツセントリックネットワーク (CCN)

- コンテンツデータごとに付与される**コンテンツ名**アドレス
  - エンドホストの位置に非依存
  - 可変長の文字列で階層的に構成
- コンテンツ名を用いた2種類のパケットによる新たな通信方式
  - Interest パケット
    - ▶ 要求者がコンテンツ名を用いてコンテンツを要求するために用いる
  - Data パケット
    - ▶ コンテンツ名と対応したコンテンツを供給するために用いる

2013/5/17 5月 IN 研究会

Osaka University

## CCN パケット転送のメリット

- マルチキャストとキャッシングによる冗長なトラフィックの削減
  - Interest の受信インターフェースを記憶し、重複した場合は集約する
  - Interest と対応する Data が返送されてきたらマルチキャストする
  - Data をルータでキャッシュする

2013/5/17 5月 IN 研究会

Osaka University

## CCN の課題

- 全く新しいネットワークアーキテクチャの実装
  - 可変長のコンテンツ名の高速処理機構
  - マルチキャストやキャッシングができる高機能なルータ
  - エンドホストの位置に非依存なコンテンツ名に基づくルーティング
- 既存の IP ネットワークとの親和性
  - 現在の IP ネットワークを CCN に即座に置き換えることは不可能
  - IP ルータと共存できる段階的な導入展開が必要不可欠

2013/5/17 5月 IN 研究会

Osaka University 7

### 研究の目的とアプローチ

- 研究の目的
  - 段階的な導入展開を考慮したCCN 実装の実現可能性の実証

→ CCN の実装と段階的な導入展開のために **OpenFlow** に着目

- アプローチ
  - OpenFlow による IP と CCN の共存
  - OpenFlow を用いて CCN を実装

2013/5/17 5月 IN 研究会

Osaka University 8

### OpenFlow の特徴

- ネットワークを自由にプログラミング可能
- OpenFlow ネットワーク上に複数の独立した仮想ネットワークスライスを実現可能

2013/5/17 5月 IN 研究会

Osaka University 9

### OpenFlow の概要

- OpenFlow コントローラとスイッチ
  - ルーティング・フォワーディング戦略と、その実現のためのハードウェアを分離
- フローエントリ単位でのネットワークプログラミング
  - フローエントリ: マッチング条件とアクションから構成

2013/5/17 5月 IN 研究会

Osaka University 10

### OpenFlow のバージョンごとの差異

バージョン	製品化	フローエントリに利用可能なプロトコル				
		MAC	TCP/UDP	IPv4	IPv6	MPLS
OpenFlow 1.0	○	○	○	◎	×	×
OpenFlow 1.3	×	◎	◎	◎	◎	○

\* アドレスやポート番号のマッチング条件で、  
 ○: 完全一致のみ指定可能  
 ◎: 最長プレフィックスマッチング(1.0)またはビットマスク機能(1.3)が指定可能

- OpenFlow 1.0
  - 現時点で最も一般的に利用されており、製品化が進んでいる
- OpenFlow 1.3
  - 最新版で機能も豊富だが、まだ実装機器が少ない

→ 既存製品を用いた早期実現性を考慮して OpenFlow 1.0 を用いる

2013/5/17 5月 IN 研究会

Osaka University 11

### OpenFlow を用いた CCN 実装の既存研究の問題

- 提案されている実装方法の具体性の欠如
  - スイッチやコントローラの役割のみ記述<sup>[1]</sup>
  - マッチング条件やアクションの仕様考察が不十分
- コンテンツ名の処理方式の欠陥<sup>[2]</sup>
  - コンテンツ名を単純にハッシュ値に変換して処理する
  - 要求時と返送時のコンテンツ名の違いに対応できない

[1] D. Syrivelis, G. Parisi, D. Trossen, P. Flegkas, V. Sourlas, T. Korakis, L. Tassiulas, "Pursuing a Software Defined Information-centric Network," Software Defined Networking (EWSN), 2012 European Workshop on, pp.103-108, 25-26 Oct. 2012  
 [2] N. Blefani-Melazzi, A. Detti, G. Mazza, G. Morabito, S. Salsano, L. Veltri, "An OpenFlow-based Testbed for Information Centric Networking," in Proceedings of Future Network & Mobile Summit 2012, 4-6 July 2012, Berlin, Germany

2013/5/17 5月 IN 研究会

Osaka University 12

### OpenFlow を用いた CCN 実装方針(1)

- コンテンツ名の高速処理
  - コンテンツ名を階層構造を保持したハッシュ値に変換
  - ハッシュ値をマッチング条件に指定できるヘッダ領域に記述
    - ▶ 最長プレフィックスマッチングが利用可能な IPv4 アドレスを利用
- マルチキャスト
  - フローエントリにポートからの送出アクションを複数記述

2013/5/17 5月 IN 研究会

Osaka University 13

### OpenFlow を用いた CCN 実装方針(2)

- キャッシング
  - OpenFlow スイッチではキャッシング不能
  - 高速なキャッシュには OpenFlow とは別のハードウェアが必要
- ルーティング
  - ルーティング機能はフォワーディング機能と独立
  - 別に行われているルーティング研究の知見を活用可能

↓

- 実現可能性を示す目的に従い、今回は簡単な実装のみ
  - キャッシングはソフトウェアで実装する
  - ルーティング情報はあらかじめ設定しておく

2013/5/17 5月 IN 研究会

Osaka University 14

### 実装方法の概要

- エンドホストにおける実装
  - ① CCN パケットの仕様
  - ② UDP 通信の改変による CCN 通信の実装
    - ▶ コンテンツ名の階層性を保持できるハッシュ変換の利用
- コントローラのプログラムの実装
  - ③ フローエントリの実装
  - ④ パケット受信時 (Packet-In イベント) 処理

2013/5/17 5月 IN 研究会

Osaka University 15

### CCN パケットの仕様

- UDP パケットのペイロードに最低限の情報を記述
  - コンテンツ名とその長さ
  - データとその長さ (Data パケットのみ)

**Interest**

name_len	name
----------	------

**Data**

name_len	name	data_len	data
2 byte	(name_len) byte	2 byte	(data_len) byte

2013/5/17 5月 IN 研究会

Osaka University 16

### UDP 通信の改変による CCN 通信の実装

- トランスポート層のポート番号を用いて Interest パケットと Data パケットを区別
- コンテンツ名をハッシュ変換して IP アドレスを決定
  - コンポーネントごとに 4 bit を割り当てる
  - ▶ 例: /text/A.txt/v1/s1
    - "text" = 2, "A.txt" = 1, "v1" = 4, "s1" = 2
    - IP アドレス: 33.66.0.0/16 (= 0x21420000)

2013/5/17 5月 IN 研究会

Osaka University 17

### コンテンツ名の階層性を保持できるハッシュ変換

- 最長プレフィックスマッチングと組み合わせると、要求時と返送時のハッシュ値の不一致を解決

コンテンツ名	IP アドレス	宛先ポート
/abc/xyz/*	33.0.0.0/8 (0x21000000)	2
/abc/xyz/text/A.txt	33.66.0.0/16 (0x21420000)	1

2013/5/17 5月 IN 研究会

Osaka University 18

### コントローラのプログラムの実装

- パケットはあらかじめ書き込まれたフローに従って高速に処理・転送される
- Interest または Data パケット受信時の Packet-In イベントに基づいてフローエントリが更新される

2013/5/17 5月 IN 研究会

Osaka University 19

## フローエントリの実装

- Interest と Data を転送する2種類のフロー
  - Interest パケット転送用フロー

<b>マッチング条件</b>	<ul style="list-style-type: none"> <li>宛先 IP アドレス = <math>X.X.X.X/X</math> [コンテンツ名のハッシュ値]</li> <li>宛先ポート番号 = <math>Port_{interest}</math> [Interest パケット用のポート番号]</li> </ul>
<b>アクション</b>	<ul style="list-style-type: none"> <li>ポートX番から送出 [あらかじめ設定したルーティング情報に従って決定]</li> </ul>

- Data パケット返送用フロー

<b>マッチング条件</b>	<ul style="list-style-type: none"> <li>宛先 IP アドレス = <math>X.X.X.X/X</math> [コンテンツ名のハッシュ値]</li> <li>宛先ポート番号 = <math>Port_{data}</math> [Data パケット用のポート番号]</li> </ul>
<b>アクション</b>	<ul style="list-style-type: none"> <li>ポートX番から送出, ポートY番から送出, ポートZ番から送出...</li> <li>[転送した Interest パケットの受信ポートに従って決定・随時追加]</li> </ul>

2013/5/17 5月 IN 研究会

Osaka University 20

## Packet-In イベント処理

- コンテンツ名ごとにパケット処理状態を記憶
  - 要求待ち状態
    - Interest パケットも Data パケットもまだ受信していない状態
  - データ待ち状態
    - Interest パケットを受信したが Data パケットがまだ来ていない状態
  - キャッシュ済み状態
    - Data パケットを受信し, キャッシュが既にある状態

2013/5/17 5月 IN 研究会

Osaka University 21

## まとめと今後の課題

- まとめ
  - CCNの実現可能性を示すために, CCNの通信機能をOpenFlowを用いて実装
  - コンテンツ名の階層構造を保持するハッシュ変換手法により既存研究の問題を解決
- 今後の課題
  - スケーラビリティの解決
    - ハッシュの衝突率
      - OpenFlow 1.3でのIPv6利用によるハッシュ空間の増大(32 bitから128 bit)
    - コントローラの状態管理の負荷
  - 不足している機能の仕様策定・実装
    - ルーティング・キャッシング
  - スイッチを用いた実機実験・評価

2013/5/17 5月 IN 研究会

Osaka University 22

2013/5/17 5月 IN 研究会