

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 〒 565-0871 大阪府吹田市山田丘 1-5

^{††} 大阪市立大学 大学院工学研究科 〒 558-8585 大阪府大阪市住吉区杉本 3-3-138

^{†††} 日本電気株式会社 クラウドシステム研究所 〒 211-8666 神奈川県川崎市中原区下沼部 1753

E-mail: [†]{a-ooka,murata}@ist.osaka-u.ac.jp, ^{††}ata@info.eng.osaka-cu.ac.jp, ^{†††}t-koide@bk.jp.nec.com,
^{††††}h-shimonishi@cd.jp.nec.com

あらまし 将来ネットワークの一つとして、コンテンツセントリックネットワーク (Content-Centric Network; CCN) が検討されている。しかし、CCN の設計は既存の枠組みに捉われない全く新しいネットワークであり、現在のインターネットから CCN へと移行するためには、段階的な展開戦略が必要だけでなく新規手法を検証するための試験用環境も十分ではない。比較的短期に実現可能な手法として OpenFlow による CCN の導入が検討されているが、これらは設計方針に関する議論はなされているものの、具体的な実装についての議論は不十分である。本稿では、最長プレフィックスマッチングなどの CCN の重要な特徴を失うことなく、名前に基づくフォワーディングとエンド間通信を実現するための、OpenFlow ネットワークの設計および実装を行う。コンテンツ名を階層的に構成されたハッシュ値へと変換する手法を新たに導入することで、OpenFlow の高速性を維持しつつ CCN が実現可能であることを明らかにし、既存研究におけるマッチング時の問題も解消する。

キーワード 新世代ネットワーク, コンテンツセントリックネットワーク (CCN), ネットワークアーキテクチャ, OpenFlow, ルーティング

A Deployment of Content-Centric Networking by Using OpenFlow Networks

Atsushi OOKA[†], Shingo ATA^{††}, Toshio KOIDE^{†††}, Hideyuki SHIMONISHI^{†††}, and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871, Japan

^{††} Graduate School of Engineering, Osaka City University
3-3-138 Sugimoto, Sumiyoshi-ku, Osaka-shi, Osaka 558-8585, Japan

^{†††} Cloud System Research Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara, Kawasaki, Kanagawa 211-8666, Japan

E-mail: [†]{a-ooka,murata}@ist.osaka-u.ac.jp, ^{††}ata@info.eng.osaka-cu.ac.jp, ^{†††}t-koide@bk.jp.nec.com,
^{††††}h-shimonishi@cd.jp.nec.com

Abstract Content-Centric Networking (CCN) is a novel architecture for dealing with various problems facing the Internet. However, there are deployment issues and the shortage of testbed environments. We focus on OpenFlow, which is a programmable environment without disrupting existing networks. Specific implementations of CCN using OpenFlow have not been examined in sufficient detail, although there have been investigations into the conceptual design. Our approach can solve the issues in existing designs by using a map from content names to hierarchically structured hash values and the longest prefix match.

Key words Future Networks, Content-Centric Networking, Architecture, OpenFlow, Routing

1. はじめに

現在、ポストインターネットとして将来ネットワークに関する研究が活発に行われているが、特にその中でインターネットの設計思想に起因する問題を解決するために、アドレス中心ではなくコンテンツ中心のネットワークとして、コンテンツセントリックネットワーク (Content-Centric Network; CCN) が注目されている [1]。最近 CCN の実現方法について具体的に検討されつつあるが、CCN には名前解決・ルーティング・キャッシング・セキュリティといった観点での実装上の課題が多く存在する [2]~[4]。

さらに、理想的には CCN はクリーンスレートからの実現が望ましいことは言うまでもないが、一方で導入コスト等を考えた場合、現在のインターネットの資産を可能な限り継承しつつ、段階的に CCN へと移行するためのシナリオについても考える必要がある。ネットワーク事業者およびエンドユーザによる CCN の自発的展開を促進するためには、CCN がインターネットと同等以上に有用であることを示し、さらに既存 IP ルータとの親和性や導入のコストなどの CCN の導入展開に関する課題も解決していく必要がある。

このような背景から、本稿では近年柔軟なトラフィック制御をプログラマビリティの導入により実現した SDN (Software Defined Network)、および OpenFlow に着目し、OpenFlow による CCN の実現手法について考える。これは OpenFlow によりパケットのヘッダ情報にもとづく柔軟な転送制御が可能となることから、CCN に適したパケットヘッダの設定を行うことでより簡便に CCN を実現することが可能であるためである。

OpenFlow を用いたコンテンツセントリックの原理に従うネットワークの実装については、これまでにもいくつか検討がなされているが、いずれも設計方針に関する議論はなされているものの [5]~[7]、CCN の具体的な実装についての議論はまだ不十分である。そこで本稿では、OpenFlow 1.0、および OpenFlow コントローラの実装である Trema を用いて、名前に基づくフォワーディングを実現する OpenFlow ネットワークの実現を目標とし、具体的なアドレッシングアーキテクチャ、およびフォワーディングエンジンの設計および実装を行う。本稿では特に名前を直接アドレスとして用いるのではなく、階層化ハッシュ関数による変換を導入することで、既存の OpenFlow ヘッダ仕様に準拠した名前アドレス表現法を実現した。これにより、OpenFlow スイッチの性能を維持したまま CCN による通信が実現できることを示す。以下、2. で OpenFlow および CCN の概要、3. で提案アーキテクチャの設計について延べ、4. でその実装手法について説明する。5. で提案アーキテクチャに関する考察を行い、最後に 6. でまとめと今後の課題について述べる。

2. CCN および OpenFlow の背景

2.1 CCN

CCN (Content Centric Network) は、コンテンツ中心の情報交換を目指して設計された、全く新しいネットワークアーキテクチャである。CCN には、次のような特徴がある [3]。

- ホスト中心ではなくコンテンツ中心
 - コンテンツごとに個別に割り当てられた名前アドレス
 - Interest パケット要求と Data パケット供給による通信
 - 優れたパケット転送を実現する戦略とハードウェア機構
- コンテンツ中心のネットワークを実現するために、すべてのコンテンツのデータが、識別子としてコンテンツ名を持つ。また、通信は端末に対する識別子である IP アドレスに代わって、コンテンツ名自体に基づいて行われる。例えば、りんごの画像には“/pictures/fruits/apple.jpg”のような名前をつけることができる。本稿では CCN アーキテクチャで示された、スラッシュを区切りとした階層型名前構造を用いることとする。

CCN における通信は、コンテンツを要求する Interest パケットと、コンテンツを供給する Data パケットによって行われる。

CCN のネットワーク内部でのパケット転送は、ルーティングプロトコルに従って Interest パケットを転送し、その転送経路を Data パケットが逆向きに辿っていくことで実現される。さらに CCN では単なるパケット転送だけではなく、重複した Interest パケットの転送の削除やマルチキャスト転送、さらに通過した Data パケットを中継ルータでキャッシングすることによる応答性能の改善などが基本機能として考えられている。CCN で用いられるルータは、それらの機能を持ったパケット転送を実現するためのデータを保持しておく 3 種類のバッファと、そのポインタを保存しておくインデックステーブルを持つ。バッファには、FIB (Forwarding Information Base) ・ PIT (Pending Interest Table) ・ CS (Content Store) の 3 種類がある。FIB は、ルーティング情報を保持しておくフォワーディングテーブルで、Interest パケットに対して次ホップ転送のためのノードの対応を行うために用いられる。PIT は、Interest パケットが到着したインターフェースを記憶しておくテーブルで、対応する Data パケットが到着したときに Data パケットを逆方向にフォワーディングするために用いられる。CS は、Data パケットをキャッシュしておくバッファで、Interest パケットが到着したときに、キャッシュ内に要求された Data パケットの存在を確認し、存在すればキャッシュされた Data パケットを送り返すことで、ネットワーク全体の伝送効率を高めることができる。インデックステーブルは、これらのバッファへのポインタを保持しておくテーブルである。

さらに、CCN ではセキュリティ性についても考慮されている。具体的には、署名・暗号鍵通信・信頼性管理・コンテンツ保護・プライバシーについて、コンテンツそのものにセキュリティを組み込んでおく内包的な仕組みが検討されている。

2.2 OpenFlow

OpenFlow とは、Software-Defined Network (SDN) の一種で、L2 スイッチや IP ルータにおける制御部分をプログラマブルにするというアーキテクチャを実現したネットワーク仕様である。ルーティングなどの経路制御ポリシーとハードウェア処理の切り離しは、OpenFlow コントローラと OpenFlow スイッチの異なる 2 つのコンポーネントによって実現される。OpenFlow コントローラがパケット経路制御ポリシーをプログラムにより実現し、OpenFlow スイッチはプログラミングされたポリシーに従ってハードウェアによる高速転送処理を実行する。コント

ローラのスイッチへの設定はフローと呼ばれる単位で行われる。フローとは、主にマッチング規則とアクションから構成され、フローのマッチング規則に書かれたパケットをスイッチが受信した場合、そのフローに書かれているアクションを実行する。

フローに設定可能なマッチング規則やアクションは、OpenFlowのバージョンによって差がある。OpenFlow 1.0の仕様は2013年1月現在多くの製品で一般的に採用されている。OpenFlow 1.0では、マッチング規則には、トランスポート層のポート番号やIPv4アドレス、MACアドレスを含むいくつかのヘッダ情報を指定することができる。IPv4アドレスに関しては、最長プレフィックスマッチングが使用可能なことも特徴の1つである。一方、2013年4月現在の最新の仕様であるOpenFlow 1.3では、マッチング規則にIPv6アドレスを用いることができるようになってだけでなく、任意のビットだけマッチングすることができるビットマスク機能も追加されている。この機能は、IPv4アドレスやIPv6アドレスだけではなく、MACアドレスも対象となっている。

3. OpenFlow を用いた CCN のアーキテクチャ

3.1 設計原理

ここでは、OpenFlow を用い CCN 通信を実現するアーキテクチャについて、その概要設計を行う。

まず、OpenFlow スイッチは従来の IP アドレス（すなわち固定長アドレス）による処理を対象としているため、可変長であるコンテンツ名をそのままアドレスとして使用することができない。そこで提案手法では、CCN で用いられるコンテンツ名を階層構造を保ったままハッシュ値へと変換する。CCN ルータは、コンテンツ名ではなく、コンテンツ名に基づいて決定されたハッシュ値を用いてパケットを送出するインターフェースを決定する。同様の既存手法 [5], [6] ではコンテンツ名の階層構造を無視したハッシュ値への変換がなされており、現在のインターネットの階層構造による集約を用いたルーティングやフォワーディングができない問題があるが、本提案手法では階層型ハッシュ値を採用しているため、ネットワークの階層構造との親和性が高いという利点がある。さらに、導入コストを削減するために、UDP 通信を OpenFlow を用いて改変する方針を取る。

本稿では、早期の実現を考慮して、OpenFlow の仕様には OpenFlow 1.0 を用い、OpenFlow の命令単位であるフローの設定方法までを詳説する。2013年4月現在、OpenFlow の最新の仕様は OpenFlow 1.3 だが、ルータやフレームワークでは対応しているものが少ない。そのため、OpenFlow 1.0 の利用を想定したアーキテクチャを記述する。しかしながら、OpenFlow 1.3 の仕様に拡張することは容易である。

3.2 パケットの仕様

CCN においてコンテンツ要求および転送のために用いられるパケットはそれぞれ Interest パケットと Data パケットの2種類である。エンドユーザは通常の UDP パケットを定められた仕様に従って生成することで、簡単に通信を行うことができる。CCN の通信を実現するために、UDP パケットが持つ UDP ポート番号、IPv4 アドレス、MAC アドレスの3つのフィールドを用い

る。ポート番号は、Interest パケットと Data パケットを区別するために用いる。説明のため、Interest パケットのポート番号は 50001、Data パケットのポート番号は 50002 と記述する。IPv4 アドレスは、コンテンツ名を変換したハッシュ値を記述するために用いる。MAC アドレスは、パケットが到着した OpenFlow スイッチのポートが、ネットワーク外部のホストと接続しているか、ネットワーク内部の OpenFlow スイッチと接続しているかを知らせるために用いる。説明のため、“11:11:11:11:11:11” はネットワーク外部を、“22:22:22:22:22:22” はネットワーク内部を意味するものとして記述する。

パケットが持つ情報は、通信の実現可能性を示すという目的に沿って極力単純化する。この UDP ペイロードの仕様を、Interest パケットと Data パケットごとに図 1 に示す。両方のパケットが持つ情報は、コンテンツ名とその長さの情報である。Data パケットはそれに加えてコンテンツのデータとその長さの情報を CCN ペイロードとして持つ。コンテンツのサイズは約 6 万バイトまでに制限されるが、CCN では、大きすぎるコンテンツデータは、下層のプロトコルで分割の必要が無い程度にまで予め分割しておくという方針を取っており、1つのデータは 1500 バイト以下に制限されるので、問題は無い。

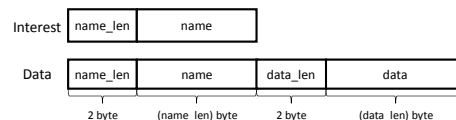


図 1 CCN パケットデータを表す UDP ペイロード

これらの情報を載せた UDP パケットを送出・受信するだけで、エンドユーザはコンテンツ転送を実現することができる。そのため、エンドユーザには専用の装置は不要である。

3.3 マッチングとフォワーディング

CCN では、各 CCN ルータにおいて、IP アドレスではなくコンテンツ名のみを用いてフォワーディングとルーティングをしなければならない。提案手法では、コンテンツ名をハッシュ値に変換し、IP アドレスのヘッダフィールドに書き込むことで、OpenFlow スイッチにおけるフォワーディング処理を実現する。

CCN ルータでは、コンテンツ名によるマッチング処理が必須となる。OpenFlow スイッチでは、UDP ペイロードをマッチング条件に指定することはできないので、コンテンツ名をハッシュ値に変換して、IPv4 ヘッダの IP アドレスのフィールドに書き込む。IPv4 アドレスの LPM(Longest Prefix Matching) 機能を利用して CCN のコンテンツ名の LPM を実現する。具体的には、コンテンツ名の構成要素ごとに 4bit ずつ割り当て、各構成要素は 1 から 15 までの値に変換する。例えば、コンテンツ名が “/pictures/apple.jpg/v1” の場合、“pictures”、“apple.jpg”、“v1” の 3 つの構成要素から成り、これがそれぞれ 1,2,3 のハッシュ値に変換されたとすると、IP アドレスは “18.48.0.0/16”(16 進数表現で 0x12300000) となる。“/pictures/apple.jpg” は IP アドレスに直すと “18.0.0.0/8” であり、このようなハッシュ値変換と IP アドレスによる LPM を組み合わせれば、コンテンツ名が

“/pictures/apple.jpg/v1”の packets とマッチするため、コンテンツ名による LPM が実現できる。

このマッチング処理に伴い、適切なインターフェースに対してパケットを送出することで、コンテンツ名のみを用いたフォワーディング処理が実現される。具体的には、表 1 のようなマッチング規則とアクションから構成されるフローをスイッチに書き込む。ただし、表 1 に加えて、すべてのフローに共通のマッチング条件として、宛先 IP アドレスフィールドがコンテンツ名を変換したハッシュ値と LPM で一致するという条件が必要である。これは、すべてのパケットの処理がコンテンツ名に基づいて行われているためである。また、コントローラによる制御を行うために、すべてのフローに共通のアクションとして、コントローラに受信したパケットを送信する処理が必要である。

アクションはそのパケットがスイッチ宛かホスト宛かによって異なる。インターフェースに受け取られずに破棄されないように、宛先アドレスを宛先ホストの値に変換する。また、ARP テーブルの混乱を防ぐために、送信元アドレスをあらかじめ決めておいたゲートウェイルータの情報に変更する (MAC アドレスは “11:11:11:11:11:11” とする)。

3.4 キャッシング

OpenFlow スイッチにはキャッシュ機構が存在していないので、キャッシングは OpenFlow コントローラが行う。コントローラで Data パケットの格納・検索を行うことで、ネットワーク内キャッシングを実装することはできる。これはソフトウェア的な実装であるため、CCN に求められるキャッシング機構の高速性には欠けるが、スイッチにキャッシュ用のストレージを接続することで、高速なハードウェア装置によって容易に拡張される。しかし、本稿では高速な専用のデバイスを開発するよりも汎用性の高いテスト環境を実現することを目的として、より柔軟性の高いコントローラによる実装を採用している。

3.5 パケット転送

パケット転送は、CCN パケットの情報を持った UDP パケットを、OpenFlow を用いてフォワーディング・キャッシングすることで実現する。ルーティング情報は FIB に予め書き込んでおくことが望ましいが、ルーティング情報が無い場合でもフラグディング処理だけを行うこととする。

図 2, 3, 4 は、提案手法における通信の例を表している。この例では、OpenFlow コントローラと OpenFlow スイッチで構成された 1 つの CCN ルータに対して、要求者 (図中で A と表記) と供給者 (図中で B と表記) の 2 人のホストが接続している。

図 2 では、要求者がコンテンツ “pict/a.jpg” を要求して通信を開始した時点の状況が示されている。要求者はコンテンツ名から宛先 IP アドレスを決定し、Interest パケットに書き込む。宛先 MAC アドレスは CCN ルータによって解決される。CCN ルータには予めコンテンツ “pict/a.jpg” に対するフォワーディング情報が OpenFlow スイッチのフローとして書き込まれており、Interest パケットは B へと送信される。

図 3 は、CCN ルータによって送出された Interest パケットが B に届く際の状況を示している。OpenFlow の機能によって IP アドレスと MAC アドレスが適切に修正されている。また、フ

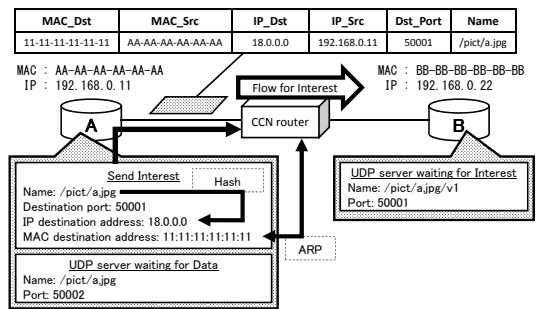


図 2 通信の概要 (1)Interest パケット送信

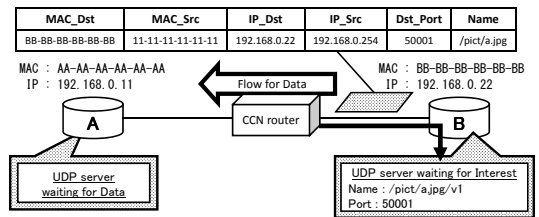


図 3 通信の概要 (2)Interest パケット受信

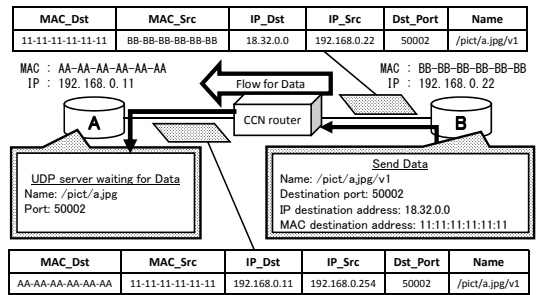


図 4 通信の概要 (3)Data パケットの送受信

ローも Interest パケット用のフローから Data パケット用のフローに書き換えられる。このフローには、サブネットマスク 255.0.0.0 として、IP アドレス 18.0.0.0 から 18.255.255.255 までのパケットがマッチする。

図 4 は、B によって返送された Data パケットが A に転送されるまでの状況を表している。B は Interest パケットへの応答として “/pict/a.jpg/v1” のコンテンツ名の Data パケットを送出する。階層構造を保ったハッシュ値への変換により、異なるコンテンツ名のフォワーディングも適切に実行される。

4. OpenFlow を用いた CCN における各コンポーネントの実装

4.1 OpenFlow スイッチへのフロー設定

OpenFlow スイッチは、OpenFlow コントローラによって設定されたフローによってパケット処理を行う。どのフローをいつ書き込むかは、コントローラが管理しているコンテンツごとの処理状態によって決定する。

スイッチのフローの設定は、コントローラが管理するコンテンツ名ごとの状態に従って行われる。状態は、そのコンテンツがどのバッファで処理されるべきかによって定義する。処理すべきバッファは、CCN における手法と同様に、コンテンツ名ごとにインデックステーブルに登録する。例えば、名前が

表1 フローのマッチング規則とアクション

目的	マッチング規則	アクション
Interest 転送 (スイッチ宛)	ポート番号が 50001	宛先 MAC アドレスを “22:22:22:22:22:22” に変更・宛先スイッチへ送信
Interest 転送 (ホスト宛)	ポート番号が 50001	宛先 IP アドレスと宛先 MAC アドレスを宛先ホストの値に変更・送信元 IP アドレスと送信元 MAC アドレスをゲートウェイルータの値に変更・宛先ホストへ送信
Data 転送 (スイッチ宛)	ポート番号が 50002	宛先 MAC アドレスを “22:22:22:22:22:22” に変更・宛先スイッチへ送信
Data 転送 (ホスト宛)	ポート番号が 50002	宛先 IP アドレスと宛先 MAC アドレスを宛先ホストの値に変更・送信元 IP アドレスと送信元 MAC アドレスをゲートウェイルータの値に変更・宛先ホストへ送信

“/pictures/apple.jpg” のコンテンツを要求する Interest パケットが初めて来たとする。このパケットは初めて来たため、CS にデータがキャッシングされていることはなく、PIT にもまだ要求が来ていない。そのため、参照すべきバッファは PIT や CS ではなく、フォワーディング情報の書き込まれている FIB である。このような状態において、インデックステーブルに “/pictures/apple.jpg” という名前に対するバッファとして FIB を登録しておけば、このパケットが FIB の情報に従って書き込まれたフローによってスイッチでフォワーディングされたことを、コントローラは即座に知ることができる。従って、コントローラは次のバッファ状態を Data パケットを待つ PIT に切り替え、適切なフローをスイッチに書き込むことができる。

状態ごとの処理は表 2 に従う。フラッディングとはパケットをすべての隣接ノードに送信することで、より厳密には、パケットを受信したポート以外のすべてのポートにパケットを送出する処理を指す。FIB エントリの登録と PIT エントリの登録・削除の処理は、コントローラが管理しているソフトウェア実装のテーブル操作と、スイッチのフローの登録・削除処理を指す。FIB エントリ登録時には Interest パケット転送用のフローを書き込み、PIT エントリ登録時には Data パケット転送用のフローを書き込む。キャッシングはソフトウェア実装のキャッシュ機構にデータを格納しておく処理を指す。Data パケット返送は、受信した Interest パケットのキャッシュヒット処理であり、格納した Data パケットをそのまま返送する。

4.2 OpenFlow コントローラの実装

OpenFlow コントローラは、1 つの OpenFlow スイッチと組み合わせることで CCN ルータとして振る舞うように実装する。そのために、スイッチに対するフローの設定やコンテンツ名ごとの状態管理に加え、ARP 解決処理と、インターフェースとホストの対応関係情報の管理を行う。

MAC アドレスは、CCN ルータが ARP 解決を行うか、ホストが ARP テーブルに予め情報を登録しておかなければならない。ARP テーブルに IP アドレスと MAC アドレスの対応を登録する手法では、大量のコンテンツを扱う場合にはそのコンテンツ名と対応する大量の IP アドレスに ARP テーブルエントリを登録しておかなければならず、煩雑である。一方、CCN ルータによる ARP 解決で MAC アドレスを取得する手法なら、MAC アドレスを持つインターフェースを経由して通信する場合には、そのインターフェースの IP アドレスと MAC アドレスさえ知っておけばよい。この ARP 解決を行う機能は OpenFlow にも Trema にも標準では備わっていないため、ARP 解決の機

能を実装している。

また、OpenFlow によって実装した CCN ルータは、ポートとホストの対応情報を管理しておかなければならない。本研究の仕様におけるパケット通信は、パケットを送受信するホスト側にとっては通常の UDP 通信と変わらない。そのため、ホスト宛にパケットを送信する場合には、適切な IP アドレスと MAC アドレスを指定しなければ、UDP ソケットにたどり着く前にインターフェースで受信拒否されてしまう。従って、CCN ルータが外部のホストにパケットを送信するために、必要なアドレス情報を管理し、外部ホスト宛のポート番号・IP アドレス・MAC アドレスを適切に指定する。

5. 考察

5.1 既存手法との比較

既存手法では、コンテンツ名を階層構造の無いハッシュ値へと動的にマッピングしている [5], [6]。しかし、Interest パケットとそれに対応する Data パケットのコンテンツ名の差異に対応するためには、階層構造を持つコンテンツ名に対する LPM が必須となる。例えば、“/pitc/a.jpg” というコンテンツ名を持つ Interest パケットと、“/pitc/a.jpg/v1/s2” というコンテンツ名を持つ Data パケットを考える。この 2 つのパケットのコンテンツ名は異なっているため、相異なる 2 つのハッシュ値が得られる。このハッシュ値が階層構造を持たない場合、CCN ルータはこの Interest パケットを転送した際に、Data パケットを返送するためのフローを記述できない。何故なら、返送されてくる Data パケットのコンテンツ名の候補は無限に存在し、階層構造を利用した集約ができないためである。Interest パケットに記述されたコンテンツ名と異なるコンテンツ名を持つ Data パケットを受信した場合にはももとのコンテンツ名を用いなければならない、ハッシュ値を用いる恩恵が得られない。

それに対して、我々の手法では、コンテンツ名を階層構造を保ったハッシュ値へと変換する。Interest パケットと全く同じではないコンテンツ名を持つ Data パケットが返送されてきたとしても、ハッシュ値の階層構造を利用して LPM による集約を行っているため、Data パケットは適切なフローにマッチして転送される。また、CCN ルータは Interest パケット通過時に Data パケット返送用のフローを書き込むことができるため、Data パケット到着時にコンテンツ名を確認しなければならない既存手法と比較して、高速に転送処理を行うことができる。

5.2 展開可能性の評価

本仕様によるネットワークは、最低限エンド間で通信を成り

表 2 OpenFlow コントローラの CCN パケットの処理

バッファ状態	Interest パケット受信時	次の状態	Data パケット受信時	次の状態
該当無し	フラッディング	PIT	FIB エントリ登録・キャッシング	CS
FIB	PIT エントリ登録	PIT	FIB エントリ登録・キャッシング	CS
PIT	PIT エントリ登録	PIT	FIB エントリ登録・キャッシング・PIT エントリ削除	CS
CS	Data パケット返送	CS	何もしない	CS

立たせる上で CCN に必要な要件を実現できている。コンテンツ名を用いたフォワーディング処理は、コンテンツ名のハッシュ化と IP アドレスフィールドを用いて実現できる。ルーティング戦略はフォワーディング戦略と独立しており、コントローラを適切にプログラミングすることによって自由に実装・拡張が可能である。CCN の特徴であるコンテンツ名の最長プレフィックスマッチングも実現されているため、最長プレフィックスマッチングを用いたルーティングも可能である。

また、ネットワーク利用者および管理者にとって、本仕様を利用することが不可能となるような障害は無く、展開可能性がある。エンドユーザにとっては設備的な負担が不要であり、アプリケーション開発も従来の UDP 通信アプリケーションを開発するのと同様に可能となる。ネットワーク管理者にとっても、既存の IP プロトコルで動くスライスと CCN プロトコルで動くスライスを共存させることができるため、移行に伴う負担を軽減することができる。

5.3 コンテンツ名を IP アドレスに変換する限界と衝突率

コンテンツ名のハッシュ値変換では、IP アドレスの 4bit ずつを 1 コンポーネントずつに割り当てる。そのため、コンテンツ名の衝突が発生する。また、IPv4 アドレスは 32bit しかないため、1 コンポーネントに 4bit 割り当てると、8 つより多くのコンポーネントから成るコンテンツ名は扱えない。

しかし、OpenFlow1.3 の仕様では、MAC アドレスや IPv6 アドレスのマッチングを扱うことができるようになるため、これを利用すれば 48bit や 128bit のハッシュ値を用いることができる。その場合には、1 コンポーネントあたりに割り当てるビット数を 4bit より大きくすることも考えられる。1 コンポーネントあたりに割り当てるビット数を B とすると、プロトコルごとの衝突確率 P_B^{protocol} は次の式で表される。

$$P_B^{\text{protocol}} = \left(1 - \prod_{i=1}^N \frac{(2^B - 1) - i}{2^B - 1} \right)^{\frac{\text{len}}{B}} \quad (1)$$

ただし、 N はコンテンツ名の数、 len はそのプロトコルのアドレスのビット長である。

図 5, 6 は、 $B = 4, 8, 16, 32$ の場合の衝突確率 P_B^{IPv4} , P_B^{MAC} および P_B^{IPv6} を示している。このグラフから、プロトコルの違いよりもパラメータ B の違いによる影響が大きいので、衝突率を下げるためには可能な限り大きな B を取ることが望ましいことが分かる。一方で、[2] によれば 99.9% の URL が 30 個以下のコンポーネントから構成されているため、コンポーネント数から見るとよりビット長の長いプロトコルで B の値を小さく取ることが望ましい。このハッシュ衝突確率とコンポーネント数のトレードオフの調査は研究課題の 1 つである。

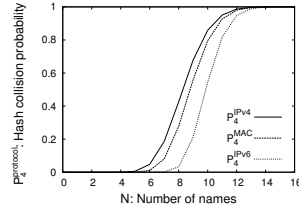


図 5 ハッシュ衝突率 ($B = 4$)

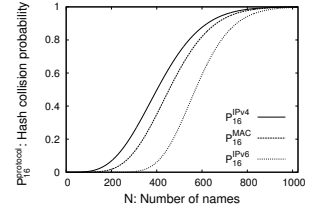


図 6 ハッシュ衝突率 ($B = 16$)

6. 結 論

インターネットの限界が認識されるにつれて、新たなネットワークとして CCN が注目を集めている。CCN には、マルチキャストやネットワーク中のキャッシングなどの多くの進展がある一方で、実現するための課題も多い。

本稿では、OpenFlow を用いた CCN のアーキテクチャと詳細な実装について述べた。コンテンツ名を階層構造を持ったハッシュ値へ変換する手法を提案し、インターネットの UDP 通信を OpenFlow によって改変することで CCN のパケット転送が実現可能であることを示した。また、既存のネットワークスライスとの共存を含む展開可能性があることも議論した。

文 献

- [1] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard, "Networking named content," 5th international conference on Emerging networking experiments and technologies, pp.1–12, Dec. 2009.
- [2] D. Perino and M. Varvello, "A reality check for Content Centric Networking," the ACM SIGCOMM workshop on Information-centric networking, pp.44–49, Aug. 2011.
- [3] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., "Named data networking (NDN) project," Technical Report NDN-0001, PARC, Oct. 2010.
- [4] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," the Re-Architecting the Internet Workshop, pp.1–6, Nov. 2010.
- [5] D. Chang, J. Suh, H. Jung, T.T. Kwon, and Y. Choi, "How to realize CDN Interconnection (CDNI) over OpenFlow?," Proceedings of the 7th International Conference on Future Internet Technologies, pp.29–30, Sept. 2012.
- [6] N.B. Melazzi, A. Detti, G. Mazza, G. Morabito, S. Salsano, and L. Veltri, "An OpenFlow-based testbed for information centric networking," Future Network Mobile Summit 2012, pp.1–9, July 2012.
- [7] I. Carvalho, F. Faria, E. Cerqueira, and A. Abelem, "ContentFlow: An introductory routing proposal for Content Centric Networks using Openflow," API, 7th Think-Tank Meeting, pp.1–2, June 2012.