4T-03

開発レイヤー分離型 OpenFlow コントローラ開発システム

着本 光大† 井口 信和†‡ 柏崎 礼生§

近畿大学理工学部情報学科† 近畿大学情報学研究所‡ 国立情報学研究所§

1. 序論

クラウドコンピューティングを支える技術の一つに仮想化技術がある．クラウドコンピューティングでは，頻繁に仮想機器の状態が変化する．それに合わせてネットワークの設定も変更する必要がある．しかし従来型のネットワーク機器は，個々に設定を管理し自律分散制御されているため，ネットワークの変化に対して柔軟に対応することが困難である．そのため，ソフトウェアでコントロールプレーンを制御し，ネットワークを集中制御する SDN(Software-Defined Networking)が利用されている [1]．

SDN を実現するためのプロトコルに OpenFlow[2] がある．OpenFlow は，コントロールプレーンである OpenFlow コントローラ(以下，コントローラ)を用いてネットワークを制御する．しかし，コントローラ開発ではネットワーク設計とアプリケーション開発の 2 つの異なる分野の知識とスキルが必要になる．ネットワークエンジニアの主な業務として，ネットワーク設計，ネットワーク構築，ネットワークの保守運用があげられる．アプリケーション開発では，プログラミング言語の知識，ライブラリ・フレームワークの知識，アジャイル開発等の開発手法の知識が求められる．そのため，アプリケーション開発の経験がないネットワークエンジニアは，コントローラ開発のためにアプリケーション開発について学習する必要がある．

そこで本研究では，ネットワークレイヤーとアプリケーションレイヤーの 2 つに開発レイヤーを分離して，コントローラ開発ができるシステム(以下，本システム)を開発した．本システムを用いることで，それぞれのレイヤーに精通した開発者が独立してコントローラ開発を進めることが可能となる．

2. 研究内容

本システムの構成を図 1 に示す．本システム

Development Layer Separated Type OpenFlow Controller Development System

Kodai TSUKIMOTO†, Nobukazu IGUCHI†‡ and Hiroki KASHIWAZAKI§

†Department of Informatics, Faculty of Science and Engineering, Kindai University

‡Cyber Informatics Research Institute, Kindai University

§National Institute of Informatics

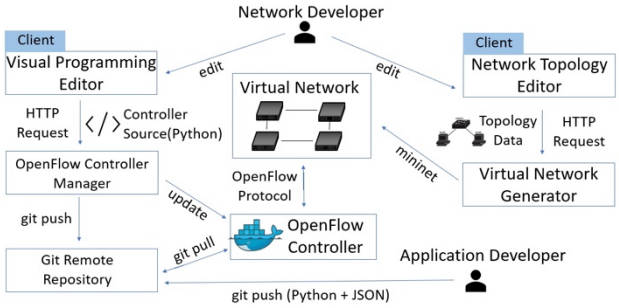

図 1 ： システム構成

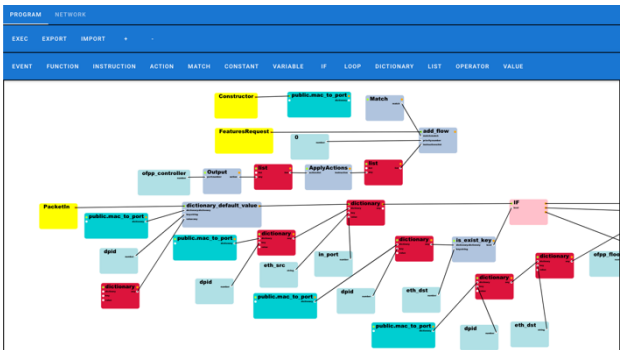

図 2 ： 本 VPE のスクリーンショット

ではネットワーク開発者は，ビジュアルプログラミング言語(Visual Programming Language，以下，VPL)を用いてネットワーク処理をプログラミングする．VPL は，プログラムの流れや機能が直感的で分かりやすい特徴を持つ．そのためテキストベースのプログラミング言語と比較して，より少ない学習時間で開発を容易に進めることができる．本システムではコントローラ開発用 VPL，VOF を新たに開発した．またネットワーク開発者は，仮想ネットワーク生成機能を用いて仮想ネットワークを構築する．アプリケーション開発者は，関数作成機能を用いてアプリケーションレイヤーを Python で開発する．

2.1. VOF

VOF の仕様について述べる．VOF は，ビジュアルプログラミングエディタ(Visual Programming Editor，以下，VPE)上で動作する．図 2 に VPE を示す．VPE は，TypeScript で実装しており Web ブラウザ上で動作する．

VOF は単一の機能を持つブロックを複数組み合わせてプログラムを構築する．ブロックは複数の接続点を持つ．接続点はブロック同士を繋げるためのインターフェースである．ブロックが持つ接続点を，フローチャートのように互いに線で結ぶことでプログラムを構築できる．接続点はフローとデータの 2 種類ある．フローはブロックの処理の順番を決める接続点である． データは値の受け渡しに使用される．データは，number，string，bool，list，dictionary 等の型を持ち，異なる型同士での接続はできないようになっている．

ブロックには条件分岐や変数を扱うブロック等様々な種類があるが，主となるブロックに Event ブロックと Function ブロックがある．Event ブロックはプログラムの起点となるブロックで，OpenFlow プロトコルの Packet In Message や Feature Message などに対応している．コントローラにパケットが送られたときに Event ブロックを起点にプログラムが実行される．Function ブロックは登録されている関数を実行する．関数はデフォルトでいくつか登録している．デフォルトで登録している関数には，パケット送信関数やフローエントリー追加関数などがある．さらに，関数作成機能を用いることで，アプリケーション開発者が自由に関数を追加できる．

VOF は，OpenFlow フレームワークの Ryu SDN Framework を用いた Python プログラムに変換され，OpenFlow コントローラとして Docker コンテナ内で実行される．

2.2. 仮想ネットワーク生成機能

本機能により，OpenFlow スイッチとホストを用いた仮想ネットワークを生成できる．仮想ネットワークの生成には，Mininet を使用する．OpenFlow スイッチには，Open vSwitch を使用する．仮想ネットワークは，ネットワークトポロジエディタ(Network Topology Editor，以下，NTE)上でスイッチとホストを結線して構築する．図 3 に NTE を示す．VPE と同様に NTE は Web ブラウザ上で動作する．

2.3. 関数作成機能

本機能により，Function ブロックで利用する関数を Python で実装できる．実装後，実装した関数のエントリーポイント，引数，戻り値を JSON 形式の設定ファイルに記述する．実装した関数と設定ファイルを，Git リモートリポジトリに push することで，Function ブロックから実装した関数の呼び出しが可能となる．

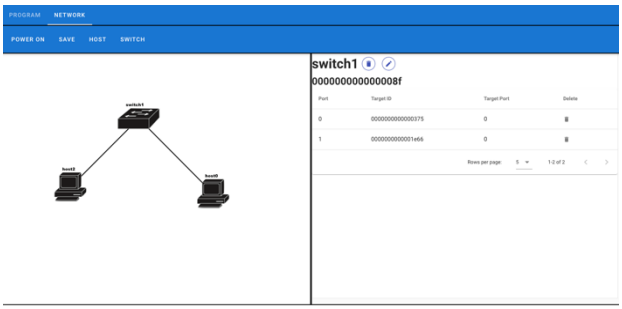

図 3 ： 本 NTE のスクリーンショット

3. 実験

本システム上で開発したコントローラで，実際にネットワークの構築が可能かを確認するために実験を実施した．実験では，Layer2 ネットワークとユーザー認証を必須とする認証ネットワークの 2 つを構築した．ネットワークの構築には，仮想ネットワーク生成機能を使用した．Layer2 ネットワークは，MAC アドレスで送信先ポートを決定するネットワークである．MAC アドレスとポートの対応付けをしてフローテーブルにフローを追加する．認証ネットワークは，接続元 IP アドレスが認証済みかを確認してから宛先に送信する．未認証の場合は，ログインサーバに転送する．認証を確認する処理のために，関数作成機能で新たに関数を作成した．

いずれのネットワークも要件通り構築できたため，本システムを用いてコントローラの開発が可能であることを確認できた．また適切にレイヤーを分離し開発が可能であることを確認した．

4. 結論

本研究では，ネットワークレイヤーとアプリケーションレイヤーの 2 つに開発レイヤーを分離して，コントローラを開発できるシステムを開発した．本システムを用いることで，ネットワークエンジニアはアプリケーション開発の学習を必要とせず，コントローラ開発を進めることができる．実験では，実際にネットワークを構築することで，開発レイヤーの分離が可能であることを確認した．

今後は，アプリケーションレイヤーの実装に用いるプログラミング言語を，Python だけではなく複数の言語に対応できるように機能の追加を予定している．

参考文献

1) M. Alsaeedi, M. M. Mohamad and A. A. Al-Roubaiey.: Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey, IEEE Access, Vol. 7, pp. 107346-107379, (2019).