

## プロジェクト紹介

# マルチネットワーク連携

2014/08/10

### ■プロジェクト名

マルチネットワーク連携

### ■プロジェクト期間

2014年8月～2015年5月

### ■プロジェクト概要

複数ネットワーク技術を連携させたE2Eネットワークの実現、OpenDaylightの導入

### ■プロジェクトの主な研究範囲

SDNコントローラ、OpenFlow、SNMP、OpenDaylight

### ■プロジェクトの目的

OpenDaylightを用いて、VPN、仮想ネットワーク、VLANといった複数のネットワーク技術を1つの統合したE2Eネットワークとして実現する。本プロジェクトでは前段階としてOpenDaylightがサポートするOpenFlowとSNMPを使用した仮想ネットワークと物理ネットワークを連携させるPOCの開発を行う。

### ■プロジェクトの背景

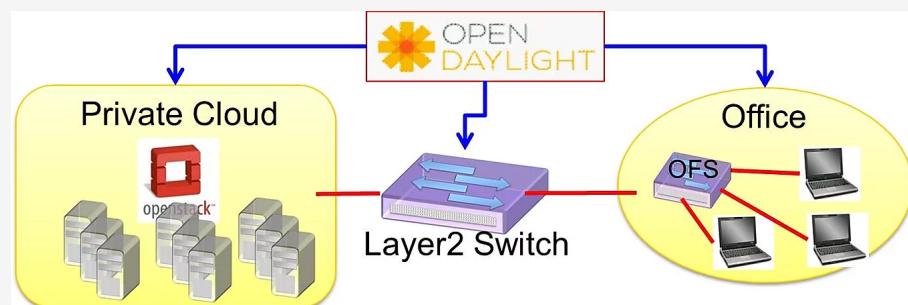
現在のSDNコントローラは多様化の兆しを見せており、定義が変わりつつある。これまでのようにOpenFlow対応スイッチのみを管理/制御対象としたものから、レガシースイッチ/ルータとの連携を意識したNetConf、LISP、BGP、SNMPなどのプロトコルを採用し、また、それらを使用したレガシーネットワークの管理/制御や、OpenStack等の仮想ネットワークとの連携、その他にもOpenFlow対応スイッチのコンフォーマンステストなど、測定器としての利用シーンなども検討されている。

そこで、沖縄オープンラボラトリではレガシーネットワークをプログラマブルに制御できるSDNコントローラでの導入を検討した。SDNコントローラの中でもレガシーネットワークを制御できるものは複数あるが、サポートするプロトコルの豊富さ、さらにOSSであり、多くのベンダーが参画していることからOpenDaylightを採用する。

### ■プロジェクト実績

#### □2014年度研究

本研究ではユースケースとしてPrivateCloud(OpenStack)とオフィス(物理ネットワーク)をL2ネットワークで接続させる。PrivateCloudのネットワークはOpenvswitch、オフィスネットワークはOpenFlow対応スイッチを用いてネットワークを構築し、その間をL2スイッチで繋ぐ構成とした。



Openvswitch-OpenFlowスイッチ間をOpenDaylightのVTN機能、L2スイッチをはさんだ、L2ネットワークをSNMP4SDN機能を使って制御し、マルチテナント型のネットワークを実現する。

※VTN:Virtual Tenant Network

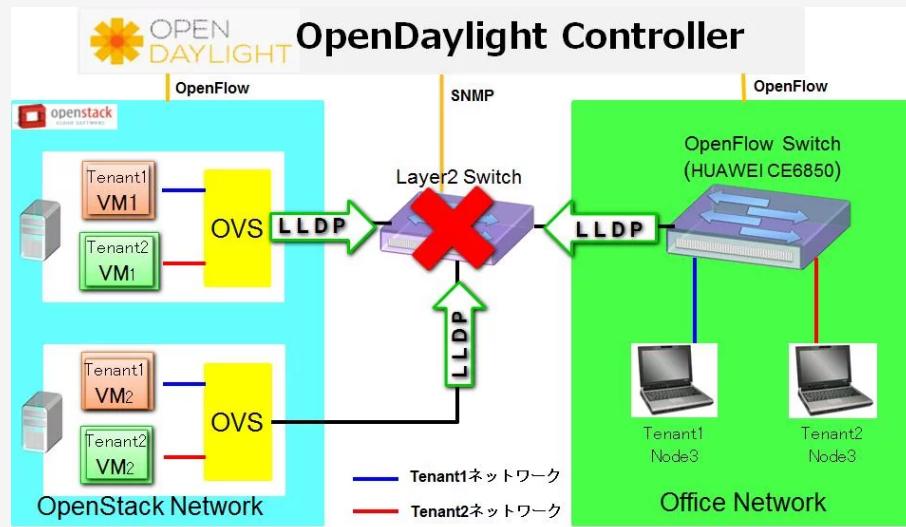
### ●検証環境 サーバ/クライアント

Node	スペック		
	OS	CPUコア数	メモリ
OpenDaylight Controller	ubuntu 12.04 LTS サーバ	2コア	4Gbyte
OpenStack Control Node	ubuntu 14.04 LTS サーバ	4コア	8Gbyte
OpenStack Compute Node	ubuntu 14.04 LTS サーバ	4コア	8Gbyte
SNMP4SDN Node1	Windows 7	2コア	4Gbyte
SNMP4SDN Node2	Windows 7	2コア	4Gbyte

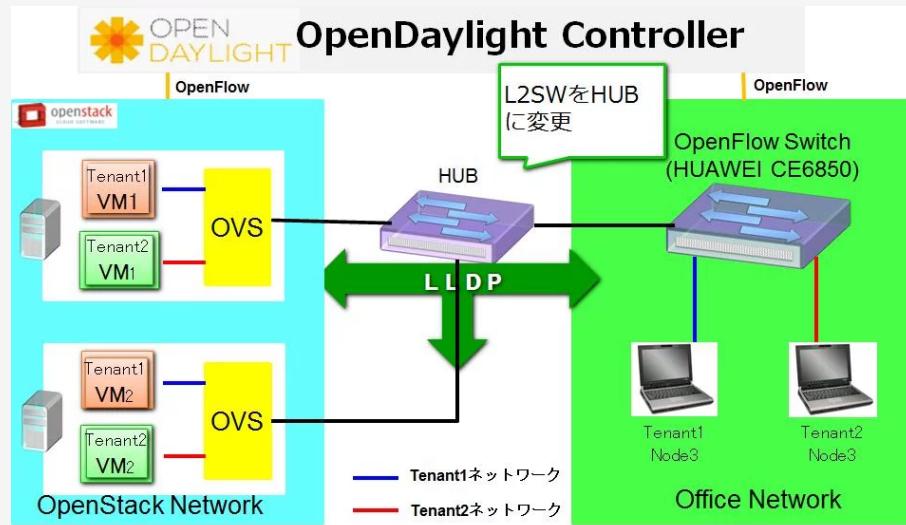
### ●検証環境 スイッチ

Switch Node	型番
OpenFlow Switch	HUAWEI CE6850
Layer 2 Switch	HP ProCurve Switch 2510
Layer 2 Switch	NEC QX-S4020P
Layer 2 Switch	Allied Telesis CentreCOM 8216XL2
Layer 2 Switch	Cisco WS-C2960-24TT-L
OVS	Open vSwitch ver 2.10

### ●検証構成 /検証結果



最初は上記の構成で検証を行った。しかしVTNで必要になるOpenFlowスイッチ間での物理トポロジを検出するためのLLDPがL2スイッチを透過しないため物理トポロジが検出できずVTNを利用することができなかった。



そこで、L2スイッチをHUBに置き換えることでLLDPによる物理トポロジを検出できるようにした。その結果、VTN機能を使用することができ、PrivateNetwork内の仮想マシンとOfficeNetwork内の物理マシンをマルチテナント型のネットワークで接続させることができた。L2スイッチをHUBに変えたのでSNMP4SDNを連携させることはできなかった。

### ●課題

- VTN機能でネットワークに接続させるためのマシン情報をPrivateCloud側はOpenStackとOpenDaylightで連携させることで自動的に登録されるが、Office側のマシン情報は手動で登録させる必要がある。
- L2スイッチに対してLLDPを透過させる必要がある。

### ●成果報告

#### 【平成26年度 沖縄オープンラボラトリ活動報告会】

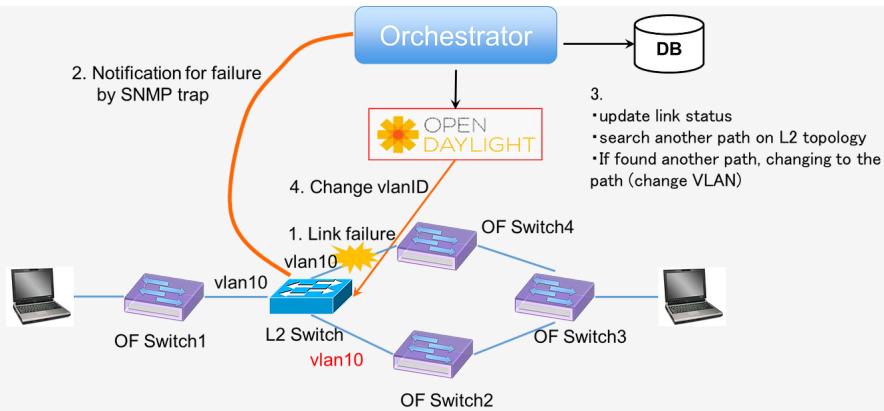
日時：平成27年2月20日（金）13：00-17：00

会場：沖縄IT津梁パーク 1F プレゼンテーションルーム(沖縄県うるま市)

#### 成果報告動画

#### 2015年度研究

2015年度は新たに以下のユースケース(High Availability)を元に検証を行った。



このユースケースは、まず両端にあるクライアント端末間で通信を行う。通信を行う際はOpenDaylightのVTNを使用し、自動的にOpenFlowスイッチにフローを登録することで通信を保障する。最初はOF Switch1とOF Switch4に接続されているL2SwitchのポートにVLAN10を設定しておく。そのため、クライアント間の通信は上のパス(OF Switch – L2 Switch – OF Switch4 – OF Switch3)を利用して通信が行われる。この状態からL2 SwitchとOF Switch間にリンク障害を発生させる。そのリンク障害はOrchestratorが検知し、Orchestrator内で新しく経路を再計算する。再計算した後に、OrchestratorはOpenDaylightに対してVLANを切り替えるためのSNMP4SDNのREST APIを実行する。VLAN切り替えの通知を受けたOpenDaylightは自身が管理する、L2 SwitchのポートをOF Switch1とOF Switch2が同一ネットワークになるようVLANを切り替える。これらの一連の処理を全てOrchestratorとOpenDaylightが自動で行い、クライアント間の通信は下のパス(OF Switch – L2 Switch – OF Switch2 -OF Switch3)に切り替わり、継続して通信を行うことができるようになる。

ただし、このユースケースを実現するには、2014年度に検出した、以下の課題を解決する必要がある。

LLDPではマルチキャストアドレスを使用しているため、Layer2スイッチがトポロジー経路上にあると、OpenFlowスイッチのトポロジーが取得できない。

この課題に対して、マルチキャストアドレスを使用せずにブロードキャストアドレスを使うことにより、正常にトポロジーを取得することに成功した。

また、本ユースケースではOrchestratorの開発を行った。Orchestratorが持つ機能を以下に示す。

- ・SNMPTrapによるLayer2Switchの障害検出
- ・ネットワーク機器および物理トポロジーの管理
- ・物理トポロジーにおける最短経路探索
- ・OpenDaylightとの連携機能

本ユースケースにおける検証環境を以下に示す。

Switch Node	型番
OpenFlow Switch	OpenvSwitch ver 2.32
Layer 2 Switch	HP ProCurve Switch 2510

Node	スペック		
	OS	CPUコア数	メモリ
OpenDaylight Controller version Lithium	ubuntu 14.04.2 LTS サーバ	2コア	4GByte
Orchestrator	ubuntu 14.04.2 LTS サーバ	2コア	4GByte
Client Node1	Windows 7	2コア	4GByte
Client Node2	Windows 7	2コア	4GByte

検証は以下の手順で実施した。

### ●【検証前準備】

各ノードへの事前設定を以下に示す。

- ・OFswitch1とOFswitch4に接続されたL2SwitchのポートをVLAN10に所属させておく
- ・OFswitch2に接続されたL2SwitchのポートをVLAN20に所属させておく
- ・L2SwitchにSNMPTrapの設定を行い、宛先はOrchestratorのサーバを指定しておく

### ●【検証手順】

1. 各Client Node間でpingによるパケット通信を永続的に行った状態にする。この状態ではまだ通信はできていないことを確認する。
2. OpenDaylightのVTN機能を使用し、仮想ネットワークを作成する。各クライアントを作成した仮想ネットワーク上に配置させる。この時点できのpingによる疎通が行えていることを確認する。
3. L2SwitchとOFswitch4の間のLANケーブルを切断させる。pingによる疎通がしばらく確認できなくなること、Orchestratorサーバ上でSNMPTrapが検出されることを確認する。
4. しばらくすると、pingによる疎通が復旧することを確認する。

本検証における成果と課題を以下に示す。

- ・レガシーとOpenFlowが混在したネットワークをプログラマブルに制御できることを見出せた。
- ・レガシーを制御するために使用したSNMPの問題点などの知見が得られた。
- ・今回のHA検証において、障害後の復旧にかかる時間が安定しなかった。

#### ●総括（成果報告）

過去の検証含め、今回の検証でもOpenDaylightの一部の機能を使用したのみであるため、今後も検証を行っていく価値があることを示した。今回の検証については2015年7月29日にアメリカで行われたOpenDaylight Summitで発表を行った。

以下にSummitの資料とセッションのURLを示す。

資料：[http://events.linuxfoundation.org/sites/events/files/slides/ODL\\_Summit.pdf](http://events.linuxfoundation.org/sites/events/files/slides/ODL_Summit.pdf)

動画：<http://y2u.be/dAlr6jDPjAk>

OpenDaylightは開発も更に活発になっており、これからも継続して検証を行っていく必要がある。また、OpenDaylightはまだ国内での実績が少ない。今後は、国内で普及させるためにOpenDaylightの機能や構造を解析し、公開していくことで、導入障壁を下げていき、利用者が使いやすくなるための活動を行っていく。