

SAC SIS 2011 チュートリアル

新ネットワーク制御技術 OpenFlow と その利用方法

2011年5月27日

NEC

システムプラットフォーム研究所

菅原智義



概要

はじめに

OpenFlowの概略

OpenFlowの仕組み

OpenFlowの利用方法

OpenFlowによるネットワーク仮想化

OpenFlowのプログラマビリティ

おわりに

はじめに

OpenFlow

- 新しいネットワークスイッチのアーキテクチャ
- ネットワークの仮想化とプログラマビリティが特徴

ネットワーク仮想化

- 物理ネットワーク上にマイ・ネットワークを構築できる

プログラマビリティ

- アプリケーションプログラムからネットワークを制御できる

『さて、あなたは何をしたいですか。』

はじめに

OpenFlowの概略

OpenFlowの仕組み

OpenFlowによるネットワーク仮想化

OpenFlowによるネットワーク仮想化

OpenFlowのプログラマビリティ

おわりに

OpenFlowとは

OpenFlowスイッチコンソーシアムが**提唱した**、ネットワーク制御技術のことである。オープンソースで開発されている。

従来のネットワーク制御方式は、主にIPアドレスのルーティングによって行われてきたが、OpenFlowではMACアドレスやIPアドレス、ポート番号などの組み合わせによって決定される一連の通信を「フロー」として定義し、フロー単位での経路制御を実現する。これによって、品質の確保やネットワークの利用率向上などが期待できるとされる。

日本国内ではNECが、OpenFlowスイッチングコンソーシアムの設立当初から参加しており、フロー単位での処理を制御する「プログラマブルフロースイッチ」の**試作・製品化**や、モバイル通信網へ応用する技術の開発などに取り組んでいる。

(IT用語辞典「バイナリ」より引用、赤字部分を一部修正)

OpenFlow.org (旧 OpenFlow Switch Consortium)

OpenFlowの情報発信の中心

- 活動の中心は、Stanford大の Prof. Nick McKeownグループ



Firefox

http://www.openflow.org/

**http://www.openflowswitch.org/
または、「openflow switch」で検索**

OpenFlow

Home Videos Documents News Research About

Innovate in Your Network
OpenFlow enables networks to evolve, by giving a remote controller the power to modify the behavior of network devices, through a well-defined "forwarding instruction set". The growing OpenFlow ecosystem now includes routers, switches, virtual switches, and access points from a range of vendors.

Learn More

Develop

Participate
The Open Networking Foundation (ONF) is now the home of the OpenFlow specification. We invite you to join the ONF and be part of the exciting standardization and commercial development and deployment of OpenFlow.

OpenFlow News

Subscribe to RSS Feed

Quick Navigation

- » OpenFlow Specs
- » Bug Tracking
- » Wiki
- » Legal
- » Log in

OpenFlow White Paper
Download the OpenFlow Whitepaper (PDF)

Internet2 building nation-wide OpenFlow Network
May 11th, 2011 by srini

Internet2, Indiana University and the Clean Slate Program at Stanford University have partnered to create a new nation-wide OpenFlow/SDN network as part of Network Development and Deployment Initiative. It would be the first OpenFlow/SDN network of this scale in wide area that would be available for both research



Open Networking Foundation

- Software-Defined Networkingの促進ための組織
- OpenFlow プロトコルの策定と普及の中心

Firefox
http://www.opennetworkingfoundation.org/ <http://www.opennetworkingfoundation.org/>

Open Networking Foundation

Welcome to the Open Networking Foundation!

The Open Networking Foundation is a nonprofit organization dedicated to promoting a new approach to networking called Software-Defined Networking (SDN). SDN allows owners and operators of networks to control and manage their networks to best serve their needs. ONF's first priority is to develop and use the OpenFlow protocol. Through simplified hardware and network management, OpenFlow seeks to increase network functionality while lowering the cost associated with operating networks.



Open Networking Foundation Announces Nine New Members and Major Presence at Interop 2011
May 6th, 2011

The Open Networking Foundation is pleased to announce that nine new members have joined since ONF's March 22 launch, and that Interop 2011 will feature an ONF

Contact Info

Email:
[Email Press Contact](#)
[Email Membership Contact](#)

OpenFlow Switch Specification

OpenFlow SwitchとControllerの間のProtocolを規定

直近のリリースは Ver 1.1.0

<http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf>



現在は、Ver.1.2の策定中

1 Introduction

This document describes the requirements of an OpenFlow Switch. We recommend that you read the latest version of the OpenFlow whitepaper before reading this specification. The whitepaper is available on the OpenFlow Consortium website (<http://openflow.org>). This specification covers the components and the basic functions of the switch, and the OpenFlow protocol to manage an OpenFlow switch from a remote controller.

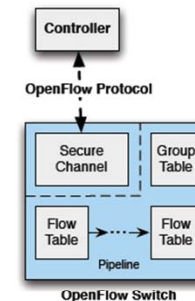


Figure 1: An OpenFlow switch communicates with a controller over a secure connection using the OpenFlow protocol.

2 Switch Components

An OpenFlow Switch consists of one or more *flow tables* and a *group table*, which perform packet lookups and forwarding, and an *OpenFlow channel* to an external controller (Figure 1). The controller manages the switch via the OpenFlow protocol. Using this protocol, the controller can add, update, and delete *flow entries*, both reactively (in response to packets) and proactively.

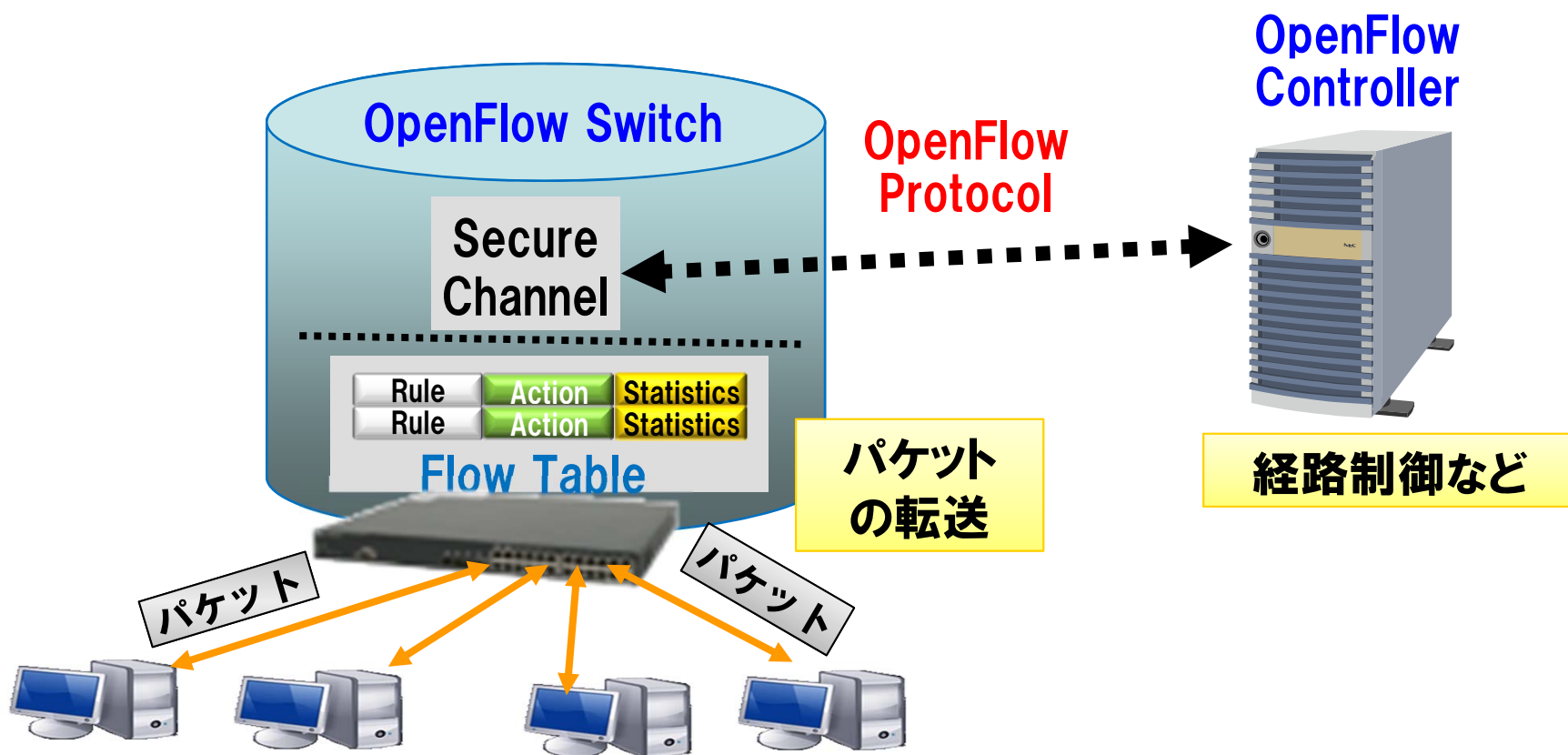
Each flow table in the switch contains a set of flow entries; each flow entry consists of *match fields*, *counters*, and a set of *instructions* to apply to matching packets (see 4.1).

Matching starts at the first flow table and may continue to additional flow tables (see 4.1.1). Flow entries match packets in priority order, with the first matching entry in each table being used (see 4.4). If a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the outcome depends on switch configuration: the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table (see 4.1.1).

Instructions associated with each flow entry describe packet forwarding, packet modification, group table processing, and pipeline processing (see 4.6). Pipeline processing instructions allow packets to be sent to subsequent tables for further processing and allow information, in the form of metadata, to be

OpenFlowの基礎知識：アーキテクチャ

- パケット転送と経路制御の機能をSwitchとControllerに分離
- 制御プロトコルはOpenFlow Protocol
- L1～L4のヘッダ情報を統合的に見て、スイッチング



OpenFlowの基礎知識：フロースイッチング

従来のネットワーク機器とは異なり、「フロー」に従い転送制御を行う

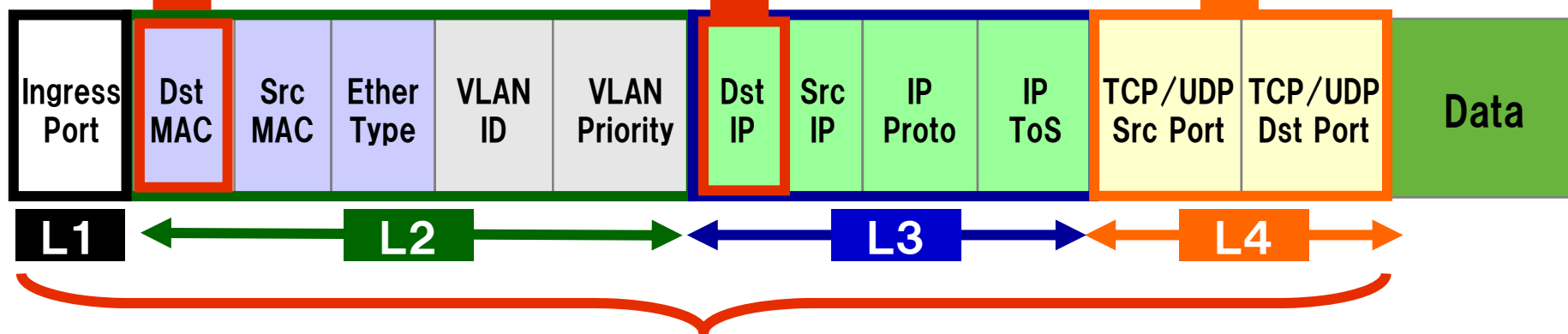
【従来のネットワーク機器】

L2/L3のレイヤーの宛先アドレスに従って個々の機器毎に転送制御

L2 (MAC) スイッチング

L3 (IP) ルーティング

(ファイアウォール等)



【フロースイッチングとは？】

フローとは、L1(物理ポート等)、L2(MAC)、L3(IP)、L4(ポート番号)の各レイヤーの任意のアドレス/識別子の組み合わせで、通信トラフィックを識別し特定するルール。そのルールの取り扱いポリシー(アクション)に従い、エンドツーエンドで最適な経路を選択して転送することをフロースイッチングと呼ぶ。

NECにおけるOpenFlowの取り組み

いち早くOpenFlow技術に注目し、試作機開発、実証実験などに取り組んできた

2008年にスタンフォード大学と共同で日米間の広帯域ネットワークを使用した実証実験

NICTなどと共同で、OpenFlow技術を用いて3D映像配信の実証実験

- 日本の14拠点に加え、韓国、香港の国際迂回回線も活用して、さっぽろ雪まつりのライブ映像および沖縄プロ野球キャンプの映像を伝送

NECのOpenFlow製品

世界で初めて、新ネットワーク制御技術「OpenFlow」に対応したネットワーク製品「UNIVERGE PFシリーズ(プログラマブルフロー)」を製品化



プログラマブルフロー・コントローラ
「UNIVERGE PF6800」



プログラマブルフロー・スイッチ
「UNIVERGE PF5240」

Interop Las Vegas 2011の Best of Interop
2011(インフラストラクチャー部門)を受賞

はじめに

OpenFlowの概略

OpenFlowの仕組み

OpenFlowの利用方法

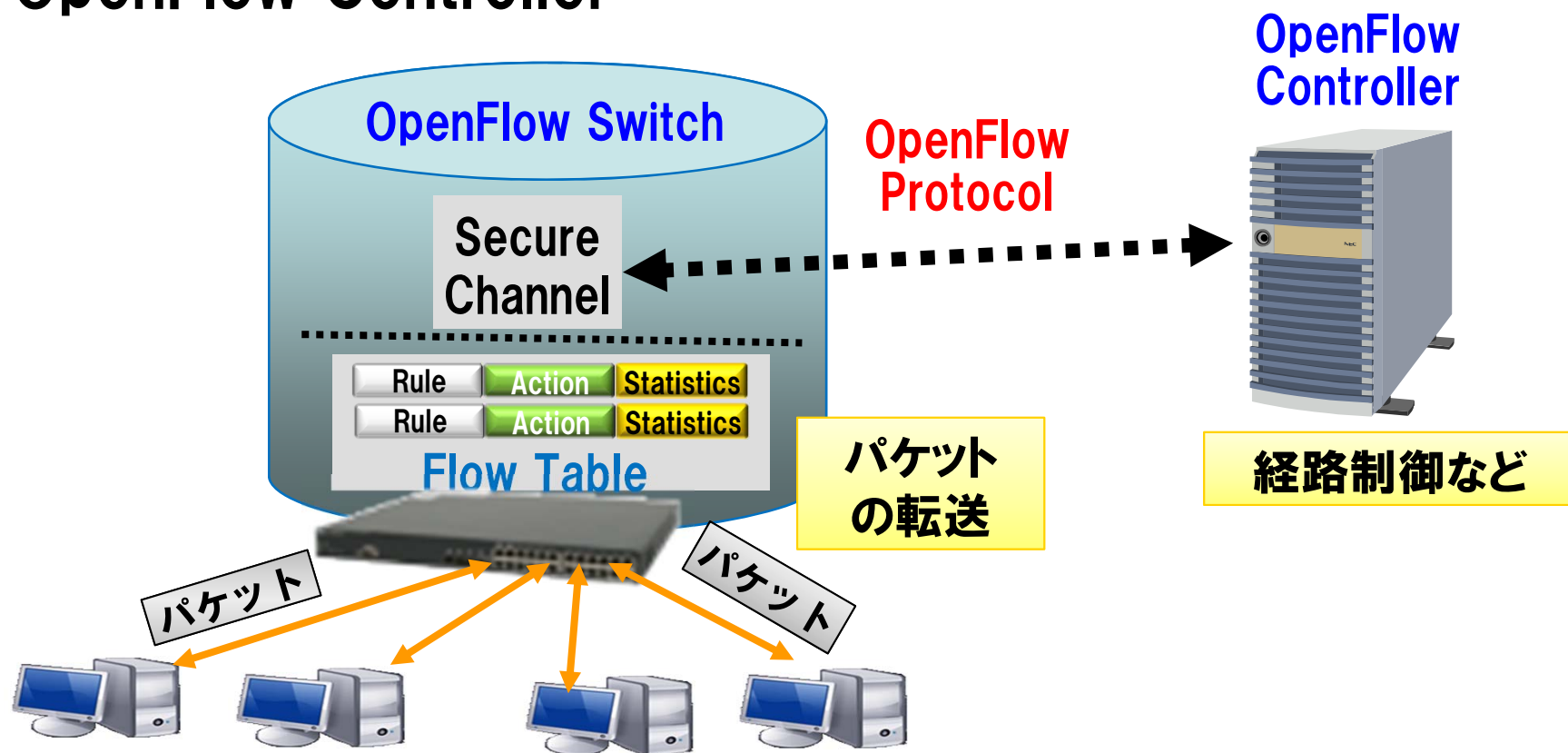
OpenFlowによるネットワーク仮想化

OpenFlowのプログラマビリティ

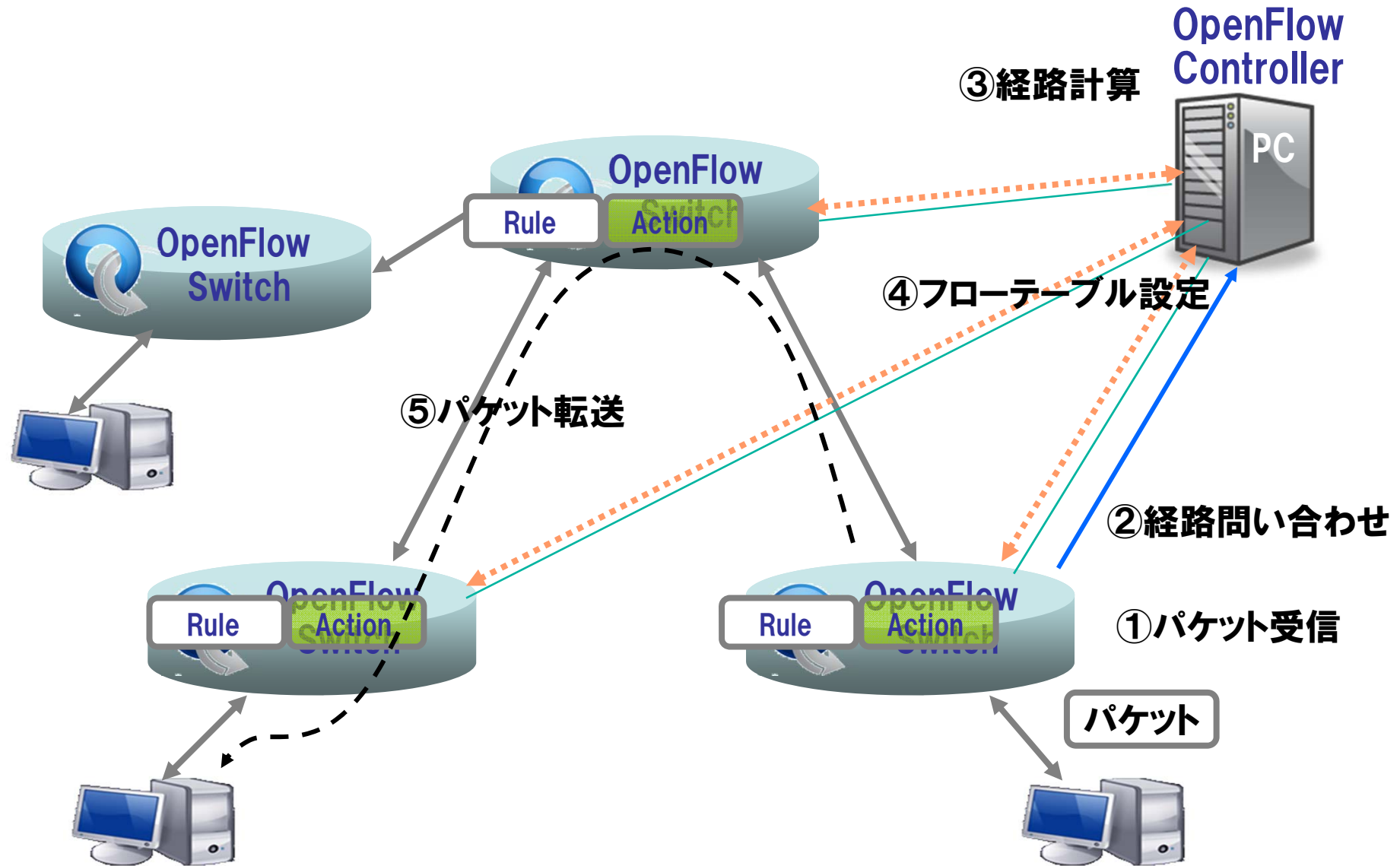
おわりに

OpenFlowの構成要素

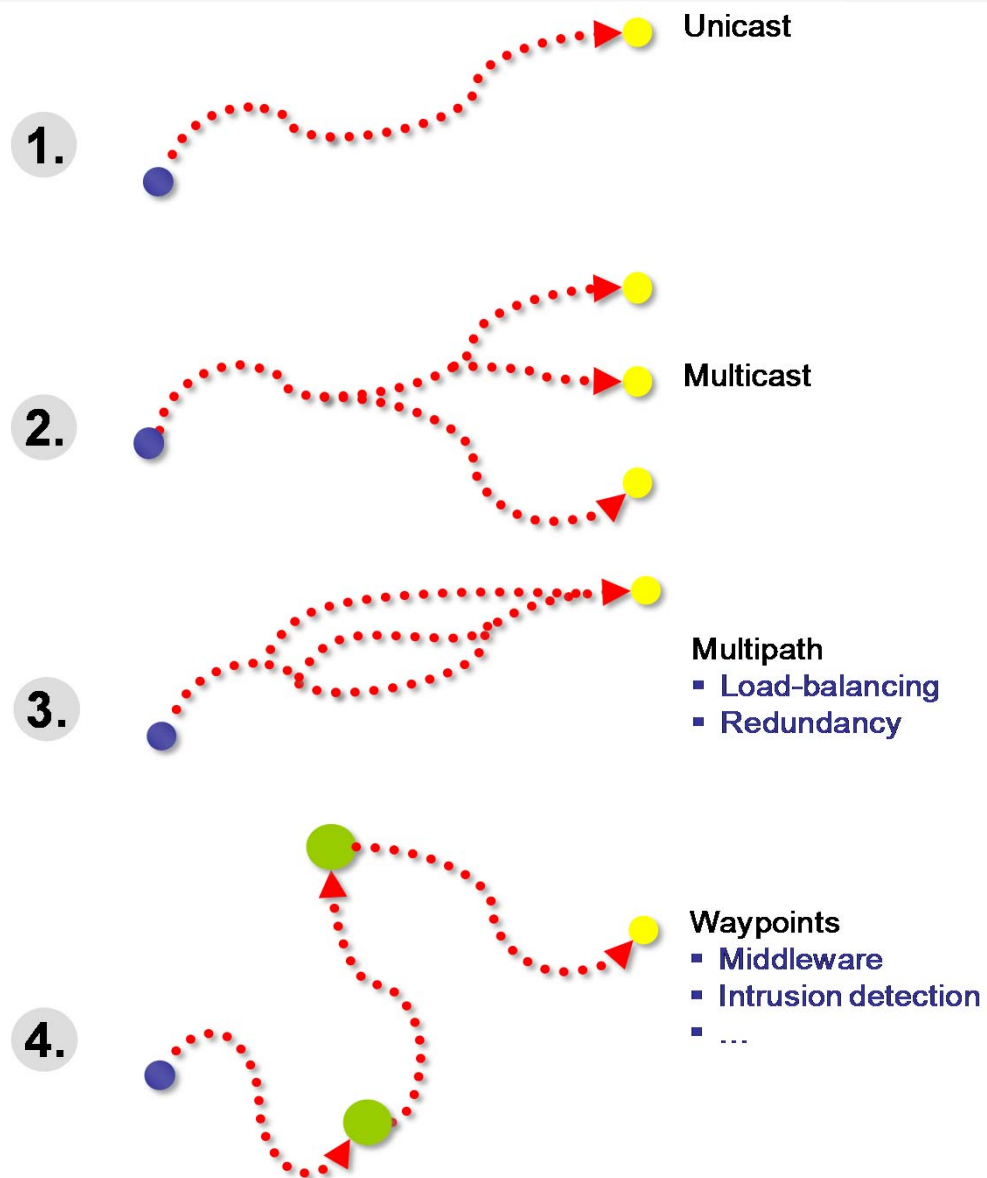
- OpenFlow Protocol
- OpenFlow Switch (Flow Tableを含む)
- OpenFlow Controller



OpenFlowの動作例



OpenFlowによる経路設定の例



OpenFlow Protocol

OpenFlow SwitchとOpenFlow Controllerの間の通信とメッセージを規定

主なメッセージの種類

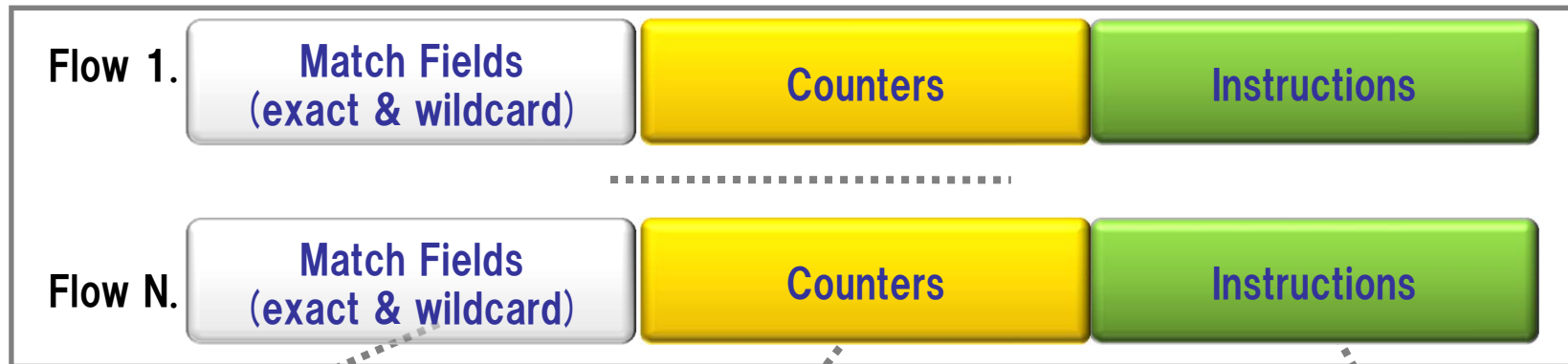
- Switchのスペック、構成、状態などの問い合わせ
- Switchの状態の報告
- Flow Tableなどの設定
- パケットの入力 (packet-in) と出力 (packet-out)
- 接続確立・死活監視

OpenFlow Protocolで用いるメッセージタイプ

```
enum ofp_type {
    /* Immutable messages. */
    OFPT_HELLO,          /* Symmetric */
    OFPT_ERROR,         /* Symmetric */
    OFPT_ECHO_REQUEST,  /* Symmetric */
    OFPT_ECHO_REPLY,    /* Symmetric */
    OFPT_EXPERIMENTER,  /* Symmetric */
    /* Switch configuration messages. */
    OFPT_FEATURES_REQUEST, /* Controller/switch */
    OFPT_FEATURES_REPLY,  /* Controller/switch */
    OFPT_GET_CONFIG_REQUEST, /* Controller/switch */
    OFPT_GET_CONFIG_REPLY, /* Controller/switch */
    OFPT_SET_CONFIG,      /* Controller/switch */
    OFPT_PACKET_IN,      /* Async */
    OFPT_FLOW_REMOVED,   /* Async */
    OFPT_PORT_STATUS,    /* Async */
    /* Controller command messages. */
    OFPT_PACKET_OUT,     /* Controller/switch */
    OFPT_FLOW_MOD,       /* Controller/switch */
    OFPT_GROUP_MOD,      /* Controller/switch */
    OFPT_PORT_MOD,       /* Controller/switch */
    OFPT_TABLE_MOD,      /* Controller/switch */

    /* Statistics messages. */
    OFPT_STATS_REQUEST,  /* Controller/switch */
    OFPT_STATS_REPLY,    /* Controller/switch */
    /* Barrier messages. */
    OFPT_BARRIER_REQUEST, /* Controller/switch */
    OFPT_BARRIER_REPLY,  /* Controller/switch */
    /* Queue Configuration messages. */
    OFPT_QUEUE_GET_CONFIG_REQUEST, /* Controller/switch */
    OFPT_QUEUE_GET_CONFIG_REPLY, /* Controller/switch */
};
```

Flow Tableの定義



柔軟なフローフィルタ定義
(例)
Port, VLAN ID, L2, L3, L4, ...

フロー統計情報
(例)
パケット数、バイト数、セッション継続時間

フローに対する処理定義
(例)
ユニキャスト, マルチキャスト, 帯域制御, 廃棄制御, 負荷分散制御, 障害回復制御, 仮想ポート (Tunnel) 転送制御,

フローフィルタ定義

(OpenFlow Switch Specification version 1.1.0 implementedより引用)

```
struct ofp_match {
    uint16_t type;           /* One of OFPMT_* */
    uint16_t length;        /* Length of ofp_match */
    uint32_t in_port;       /* Input switch port. */
    uint32_t wildcards;     /* Wildcard fields. */
    uint8_t dl_src [OFP_ETH_ALEN]; /* Ethernet source address. */
    uint8_t dl_src_mask [OFP_ETH_ALEN]; /* Ethernet source address mask. */
    uint8_t dl_dst [OFP_ETH_ALEN]; /* Ethernet destination address. */
    uint8_t dl_dst_mask [OFP_ETH_ALEN]; /* Ethernet destination address mask. */
    uint16_t dl_vlan;       /* Input VLAN id. */
    uint8_t dl_vlan_pcp;    /* Input VLAN priority. */
    uint8_t pad1 [1];      /* Align to 32-bits */
    uint16_t dl_type;       /* Ethernet frame type. */
    uint8_t nw_tos;         /* IP ToS (actually DSCP field, 6 bits). */
    uint8_t nw_proto;       /* IP protocol or lower 8 bits of* ARP opcode. */
    uint32_t nw_src;        /* IP source address. */
    uint32_t nw_src_mask;  /* IP source address mask. */
    uint32_t nw_dst;       /* IP destination address. */
    uint32_t nw_dst_mask;  /* IP destination address mask. */
    uint16_t tp_src;        /* TCP/UDP/SCTP source port. */
    uint16_t tp_dst;       /* TCP/UDP/SCTP destination port. */
    uint32_t mpls_label;    /* MPLS label. */
    uint8_t mpls_tc;        /* MPLS TC. */
    uint8_t pad2 [3];      /* Align to 64-bits */
    uint64_t metadata;     /* Metadata passed between tables. */
    uint64_t metadata_mask; /* Mask for metadata. */
};
```

フローに対する処理定義

Output（必須）

- 指定した物理ポートにパケットを転送する

Drop（必須）

- 何もアクションしない（パケットは捨てられる）

Group（必須）

- 指定したグループを通してパケットを処理する

Push-Tag/Pop-Tag

- VLANタグやMPLSタグを付加・削除する

Set-Queue

- パケットにキューIDを付加する（QoS制御のために使用）

Set-Field

- パケットのフィールドを変更する

フロー統計情報

下記は、Counter名とサイズ(bit)を表す

Per Table

Reference count (active entries) 32

Packet Lookups 64

Packet Matches 64

Per Flow

Received Packets 64

Received Bytes 64

Duration (seconds) 32

Duration (nanoseconds) 32

Per Port

Received Packets 64

Transmitted Packets 64

Received Bytes 64

Transmitted Bytes 64

Receive Drops 64

Transmit Drops 64

Receive Errors 64

Transmit Errors 64

Receive Frame Alignment Errors 64

Receive Overrun Errors 64

Receive CRC Errors 64

Collisions 64

Per Queue

Transmit Packets 64

Transmit Bytes 64

Transmit Overrun Errors 64

Per Group

Reference Count (flow entries) 32

Packet Count 64

Byte Count 64

Per Bucket

Packet Count 64

Byte Count 64

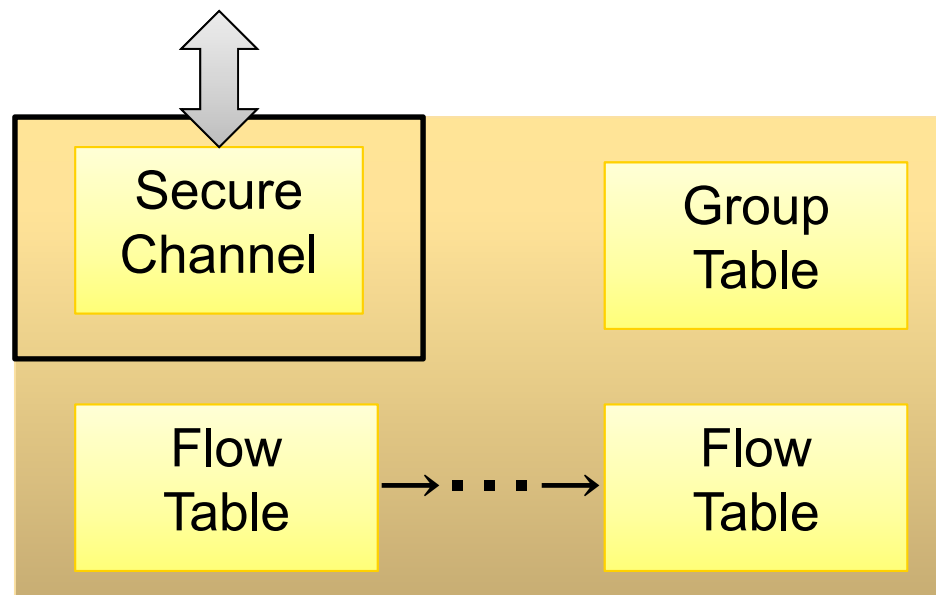
OpenFlow Switch

1つ以上の flow table と 1つの group table

- パケットの検索、転送、統計に利用

Secure channel

- OpenFlow Controllerとの通信用



OpenFlowスイッチの種類

L2スイッチベース

- 最も一般的なOpenFlowスイッチの形態
- 既存のL2スイッチをベースにファームウェアの改造によってOpenFlow機能を持たせたものが多い
- L2～L4のヘッダを処理
- NEC, HP, Juniper, Quanta, Arista等

ソフトウェアスイッチ

- OpenFlowスイッチをソフトウェアとしてホストに内蔵したもの
- Open vSwitch, Stanford Reference Switch等

トランスポートノードベース

- トランスポートノードのパス設定をOpenFlowプロトコルで操作可能としたもの
- Cienna, Fujitsu America等

ワイヤレス

- WiFi-APやWiMAX-BS/ASNGWにOpenFlow機能を付加したもの

OpenFlow Controller

■ PCサーバなどで実行されるプログラム

- 経路制御、トポロジー管理などの上位のネットワーク制御を行う

■ オープンソース

- NOX 最も歴史が古く、利用者も多いコントローラ
- SNAC NOXの改良版でGUIが特徴
- Beacon Java版のController
- Trema OpenFlow Controllerのフレームワーク

■ 製品

- プログラマブルフロー・コントローラ（NEC）

Trema

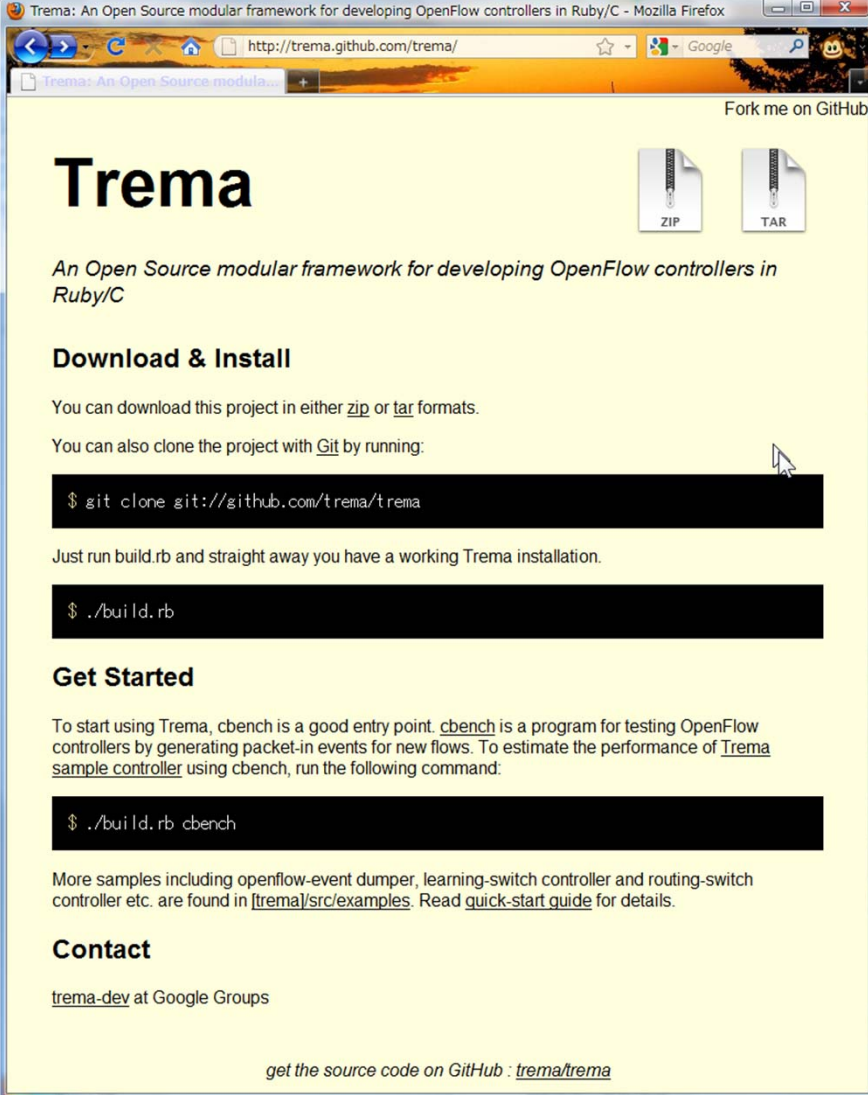
■ NECが研究開発した、OpenFlow Controller用ソフトウェアプラットフォームをオープンソースとして、githubに公開
<http://trema.github.com/trema/>

■ プラットフォームであり、経路制御などは上位モジュールとして、利用者(開発者)が記述する

- サンプルコードは付属する

■ RubyとCで上位モジュールを記述できる

■ ライセンスは、GPL ver.2



Trema: An Open Source modular framework for developing OpenFlow controllers in Ruby/C - Mozilla Firefox

<http://trema.github.com/trema/>

Trema: An Open Source modular framework for developing OpenFlow controllers in Ruby/C

Fork me on GitHub

Trema

An Open Source modular framework for developing OpenFlow controllers in Ruby/C

Download & Install

You can download this project in either [zip](#) or [tar](#) formats.

You can also clone the project with [Git](#) by running:

```
$ git clone git://github.com/trema/trema
```

Just run build.rb and straight away you have a working Trema installation.

```
$ ./build.rb
```

Get Started

To start using Trema, [cbench](#) is a good entry point. [cbench](#) is a program for testing OpenFlow controllers by generating packet-in events for new flows. To estimate the performance of [Trema sample controller](#) using [cbench](#), run the following command:

```
$ ./build.rb cbench
```

More samples including openflow-event dumper, learning-switch controller and routing-switch controller etc. are found in [\[trema\]/src/examples](#). Read [quick-start guide](#) for details.

Contact

[trema-dev](#) at Google Groups

get the source code on GitHub : [trema/trema](#)

はじめに

OpenFlowの概略

OpenFlowの仕組み

OpenFlowの利用方法

OpenFlowによるネットワーク仮想化

OpenFlowのプログラマビリティ

おわりに

OpenFlowを何に使うか

新しいネットワーク制御の試作・実験

- 経路制御やQoS制御など上位のネットワーク制御を自分で実装する
- パケットスイッチなど下位のネットワーク制御は実装しなくてよい
- すでに、学術実験網での利用が始まっている

仮想ネットワークの実現

- VLANのように数の制約を受けずに、自由に仮想的なネットワークを構築する
- すでに、プログラマブルフロー・コントローラが仮想ネットワーク機能を提供

ITとネットワークの統合制御の実現

- プログラムの要求に合わせてネットワークを制御する
- HPC分野などでの利用拡大が予想される

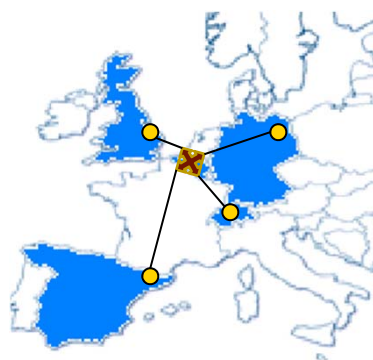
OpenFlowの学術実験網での利用

仮想化／(ユーザー)プログラマブルネットワークの実現

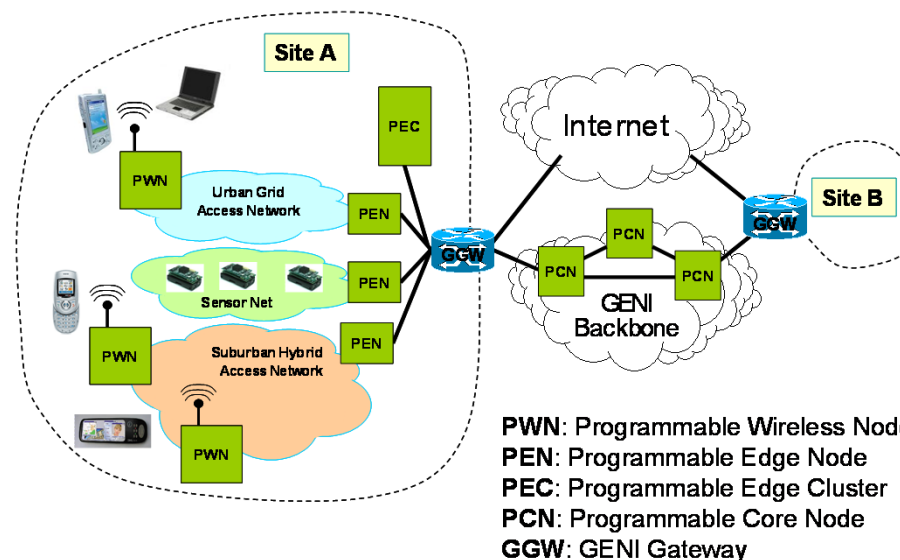
- Computer & network integrated programmable network node
- Optical dynamic circuit switch
- Programmable radio & dynamic mobility

OpenFlowテストベットに関する各国の取り組み

- 米国 NSF-GENI
- EU FP7-OFELIA
- 日本-JGN2plus/JGN-X



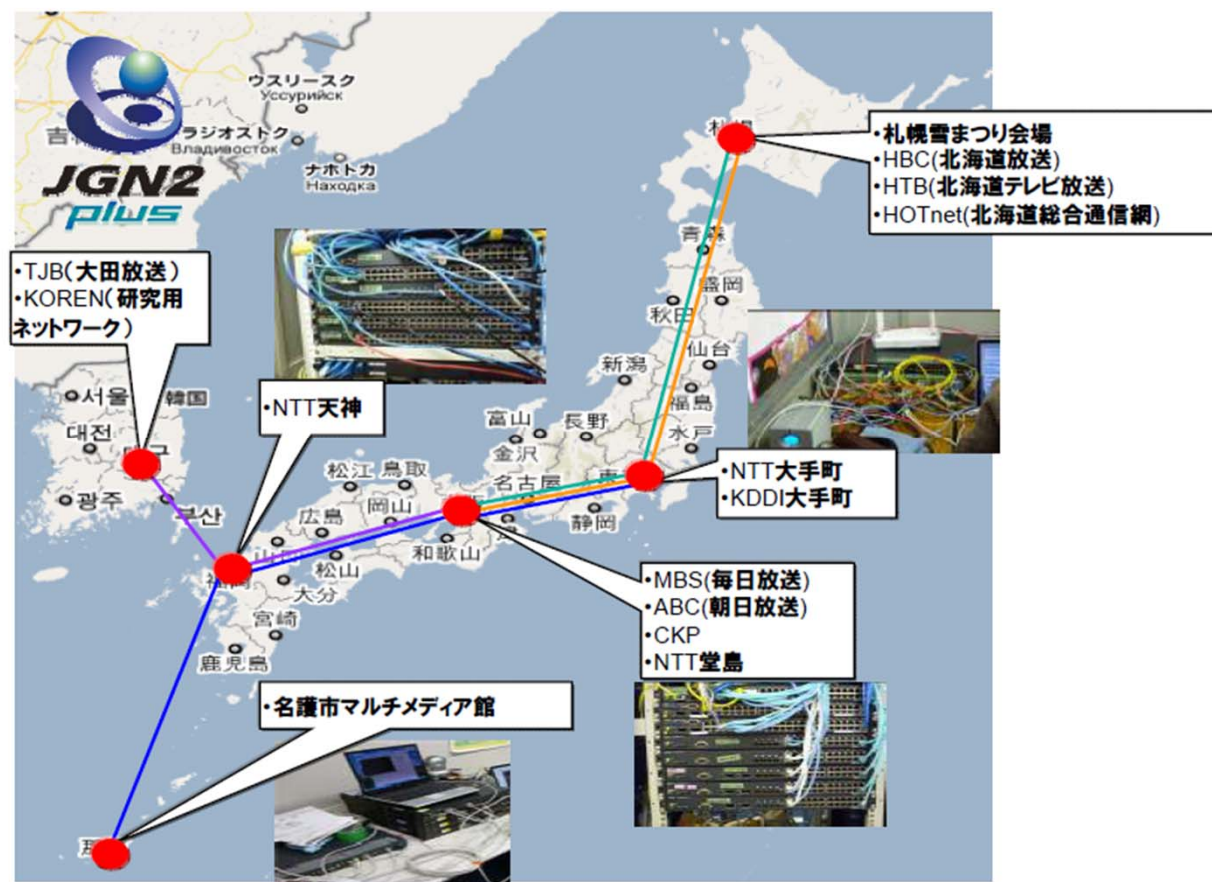
EU: FP7-OFELIA Project



USA: NSF GENI Project

JGN2plus上での映像伝送実証実験

- NECとNICTの共同でJGN2Plus上にOpenFlow通信網を構築(2010年3月)
- 広域映像伝送の実証実験を実施し、さっぽろ雪まつり・沖縄プロ野球キャンプ映像伝送に成功



はじめに

OpenFlowの概略

OpenFlowの仕組み

OpenFlowの利用方法

OpenFlowによるネットワーク仮想化

OpenFlowのプログラマビリティ

おわりに

仮想化とは

■ 抽象化(abstraction)

- エミュレータ、シミュレータ
- オペレーティングシステム(プロセス、ファイル、ソケット)
- Javaなどの仮想マシン

■ 多重化(multiplexing)

- VMwareなどの仮想マシン(VM)
- ハードディスクなどのパーティション

■ 集約化(aggregation)

- グリッド
- RAID、論理ボリュームマネージャ(LVM)

ネットワーク仮想化とは

■ 抽象化（abstraction）

- Virtual Private Network（VPN）
- ネットワークアドレス変換（NAT）

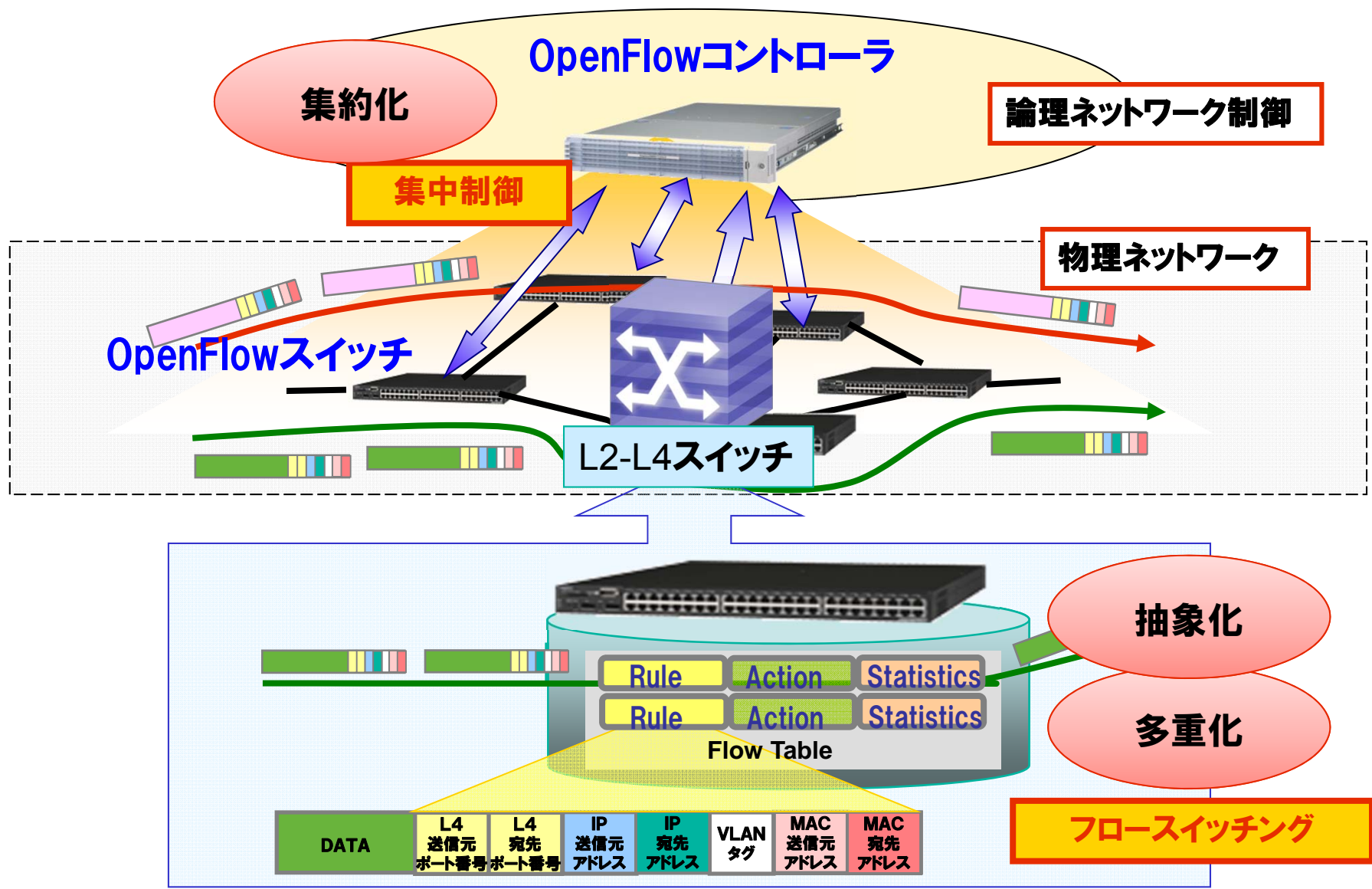
■ 多重化（multiplexing）

- VLAN（Virtual LAN）
- 単一スイッチの仮想多重化

■ 集約化（aggregation）

- 複数スイッチの統合仮想化
- チャネルボンディング

OpenFlowによるネットワーク仮想化



OpenFlowによるネットワーク仮想化の課題

■ 端末 & サーバーのモビリティ実現

- 端末が動く、サーバも動く
- アドレス/サブネットの仮想化

■ ネットワークリソースの利用効率向上

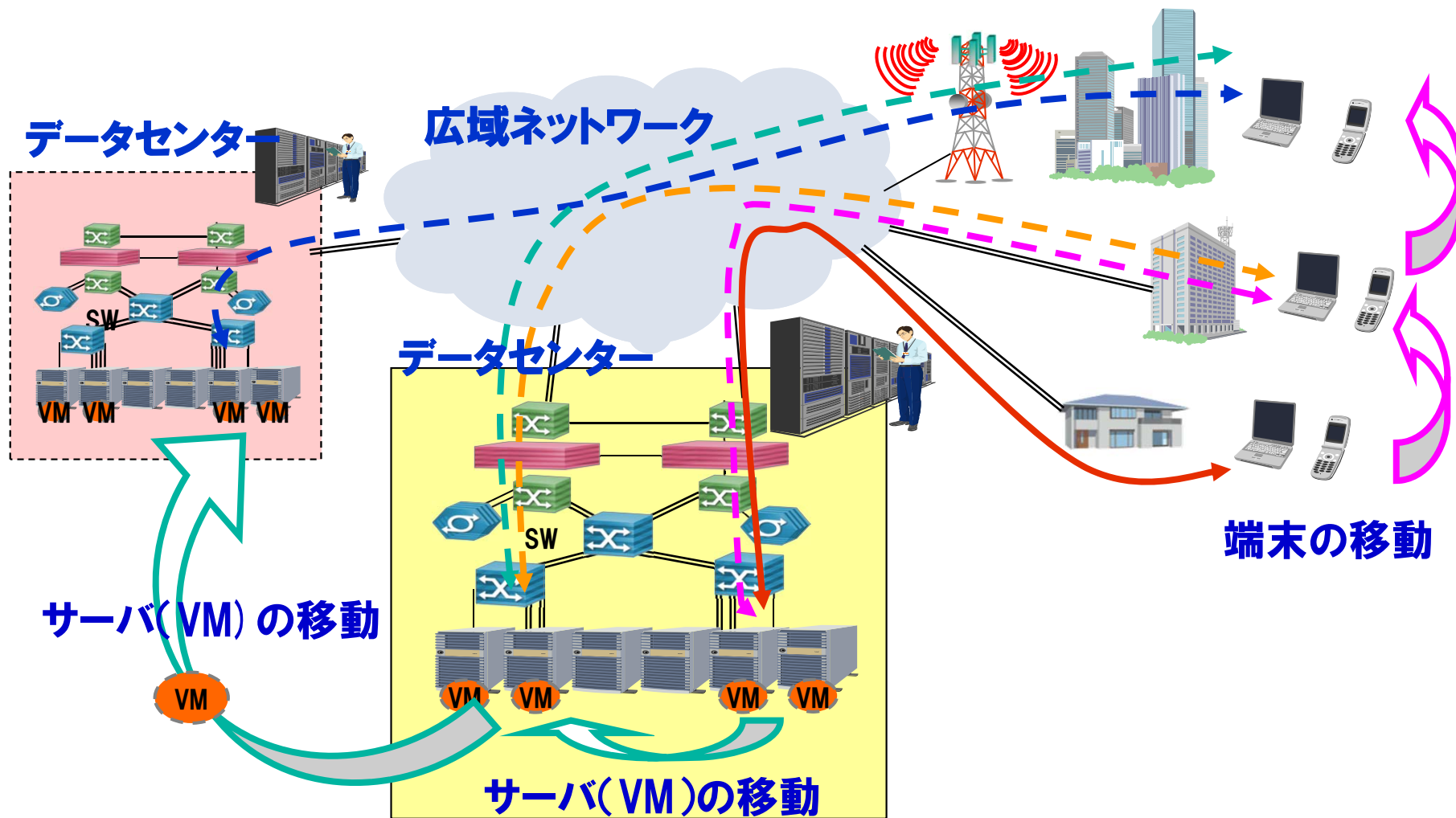
- シームレスなスケールアウト
- ネットワークリソースの共有
- 部分的なリソースの休止

■ サーバーとネットワークの一括管理

- 仮想スイッチ(vSwitch)とネットワークスイッチの連携
- サーバー & ネットワークの一括運用監視制御

端末 & サーバーのモビリティ実現

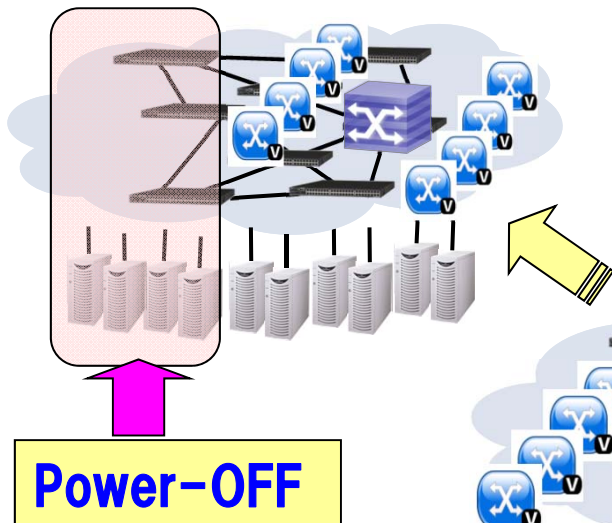
■ 端末やサーバーの移動に合わせて、フローを追従させる



ネットワークリソースの利用効率向上

ネットワーク・スケールアウト

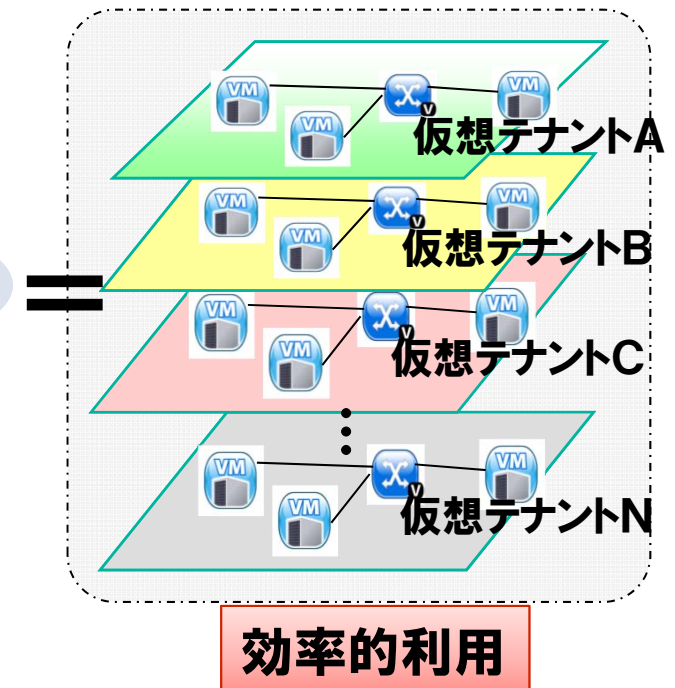
部分的なリソース
の休止



省電力

スケーラビリティ

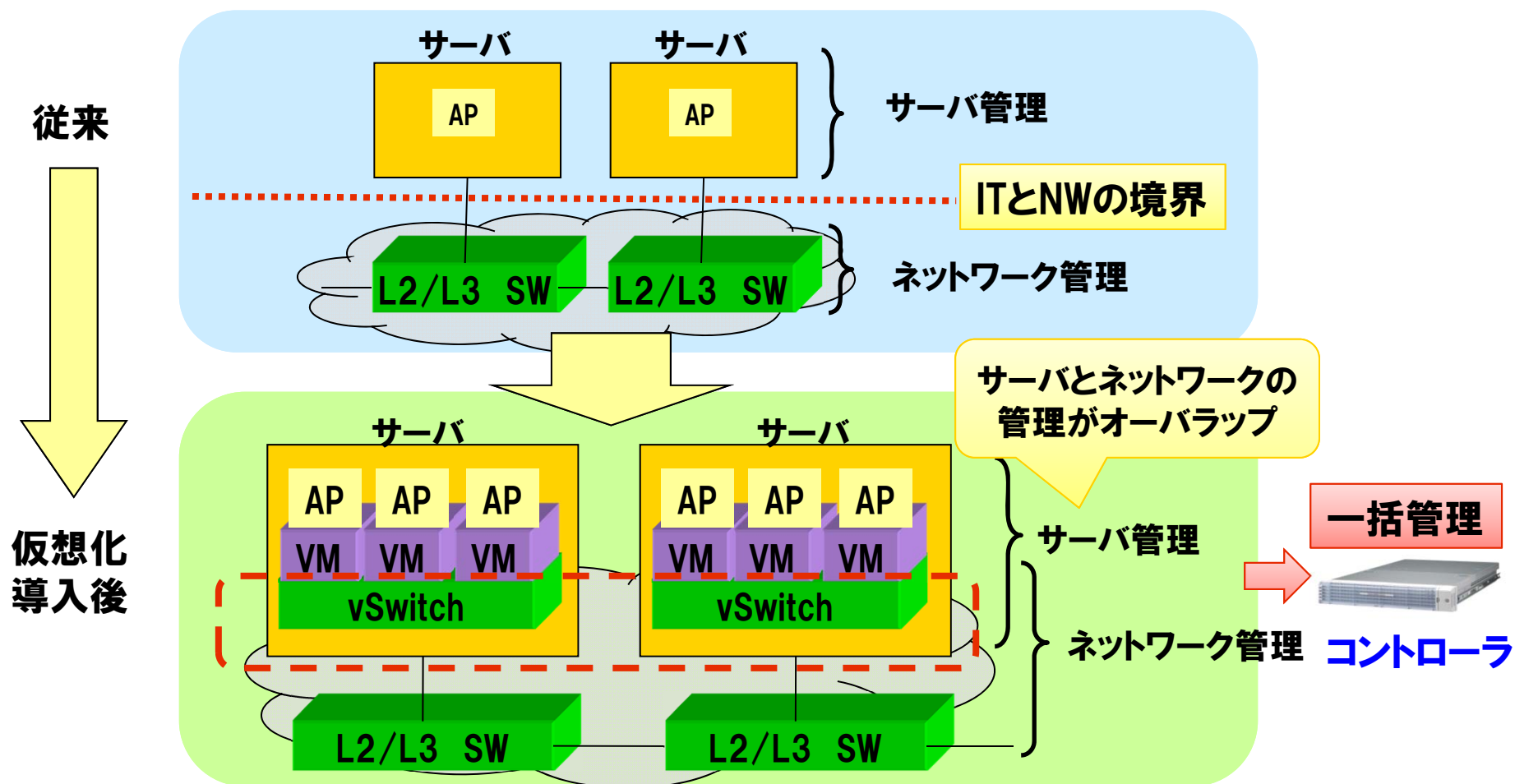
ネットワークリソースの共有



サーバーとネットワークの一括管理

従来と異なりサーバとネットワークの境界線が曖昧なものになりつつある。

→ OpenFlowコントローラによるサーバとネットワークの一括管理



はじめに

OpenFlowの概略

OpenFlowの仕組み

OpenFlowの利用方法

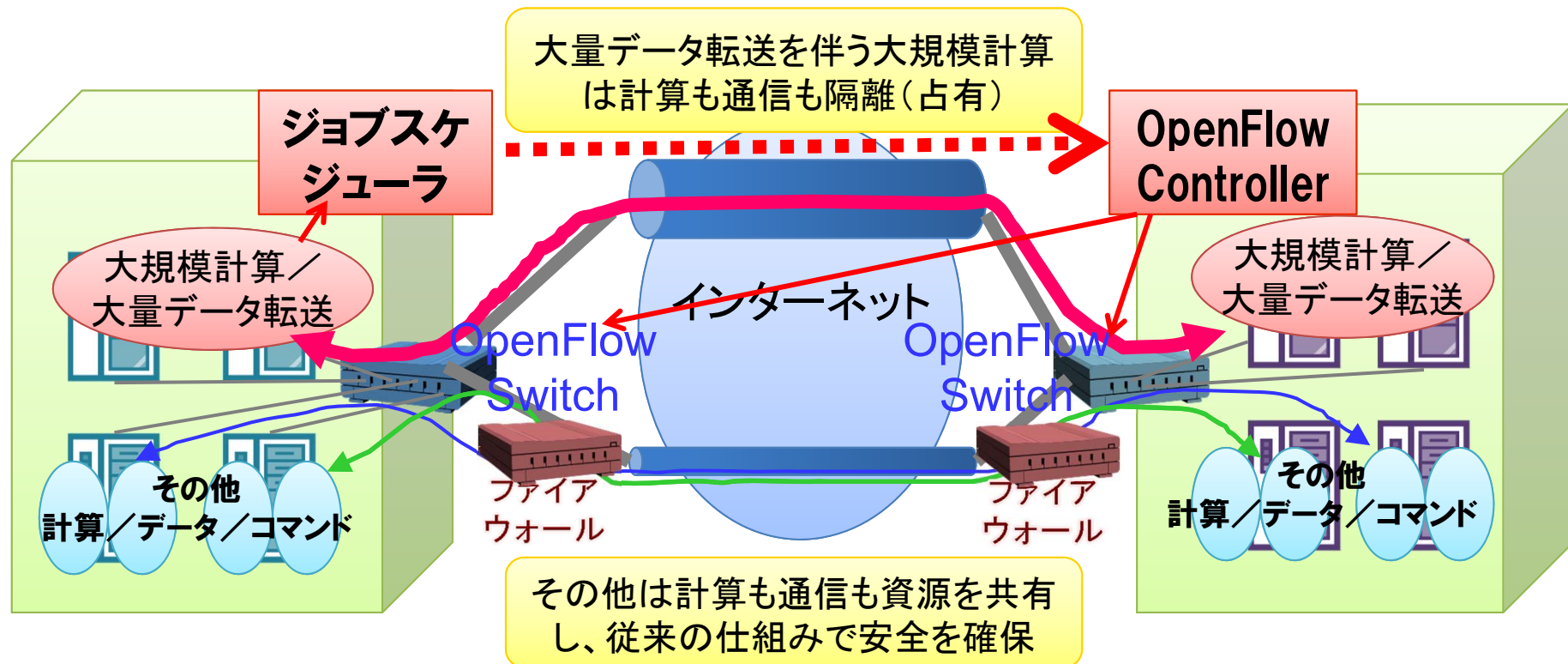
OpenFlowによるネットワーク仮想化

OpenFlowのプログラマビリティ

おわりに

ジョブスケジューラ連携制御

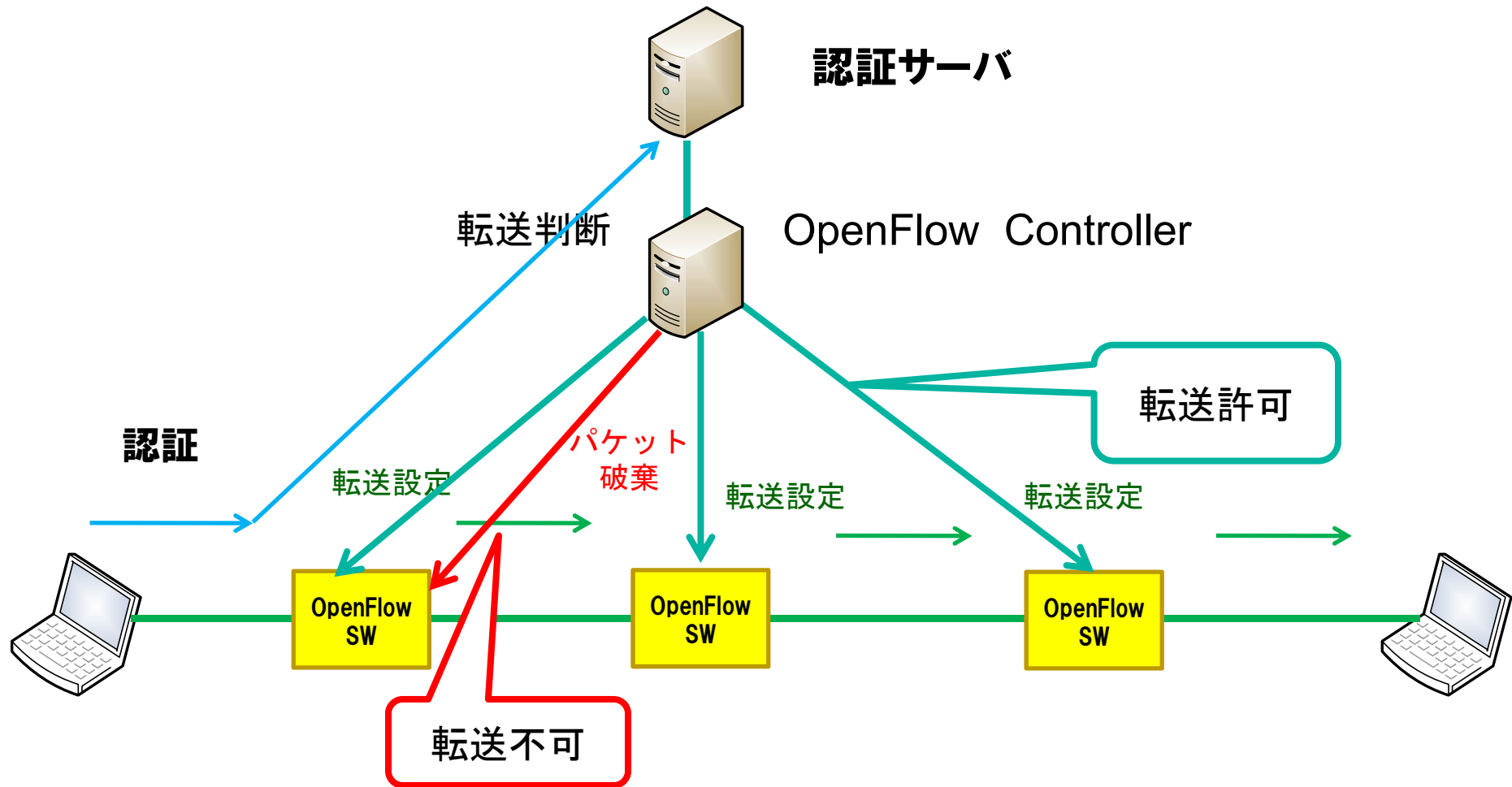
ジョブスケジューラの指示により、適切な経路設定をOpenFlow Switch に指示し、ジョブが要求する通信性能やセキュリティを確保



- ① 投入されるジョブの特性に応じて、適切な経路(フロー)を選択
- ② ファイアウォールなしでも、優先される経路(フロー)を隔離

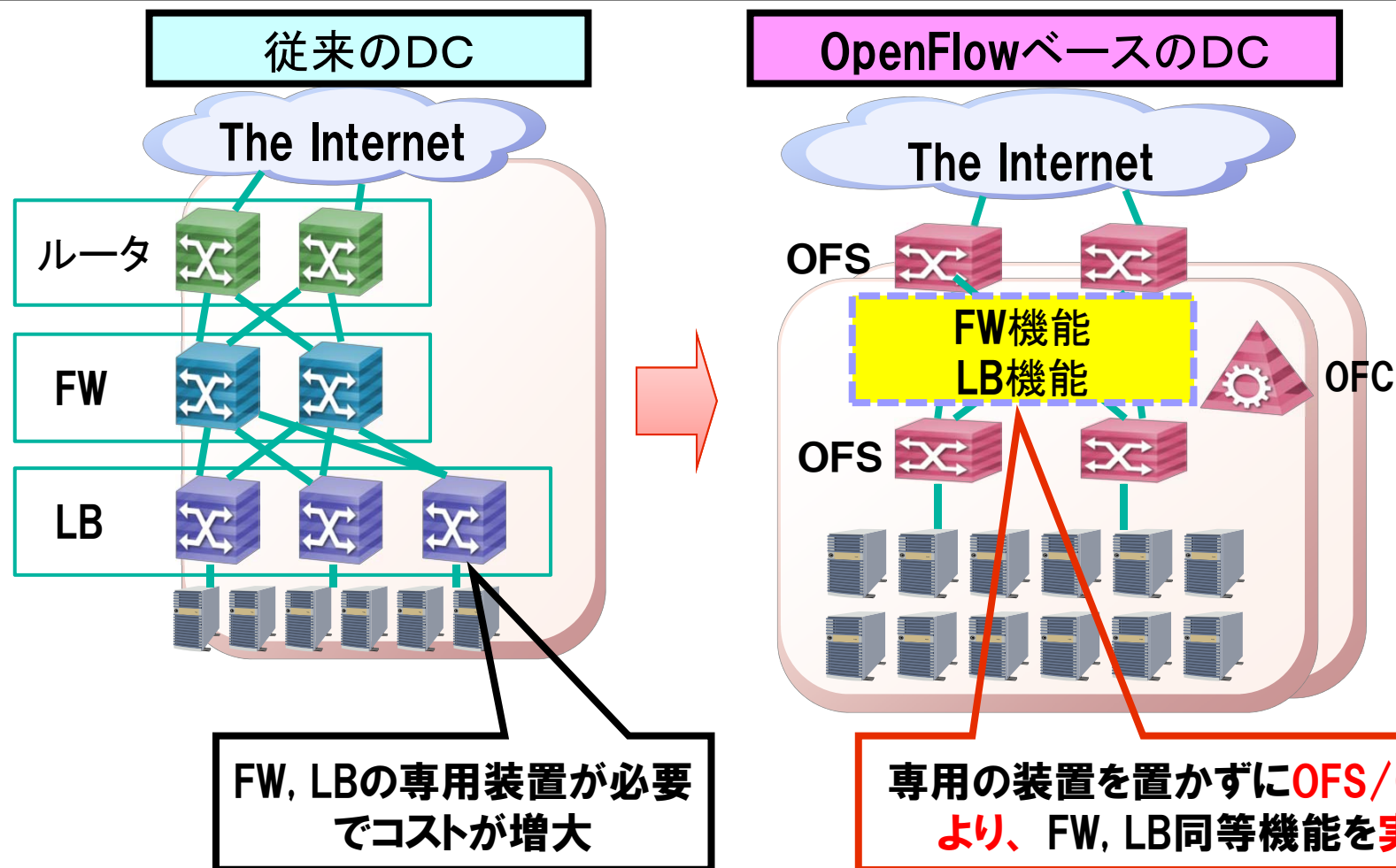
ユーザ(端末)認証連携制御

ユーザ(端末)認証の結果に基づいて、経路設定の可否を判断



OpenFlowベースのファイアウォール／ロードバランサ

- ファイアウォール(FW)／ロードバランサ(LB)等の機能をOpenFlowで実現
 - サーバを追加してもFW/LBの再配置、接続構成変更が不要



おわりに

■ 新ネットワーク制御技術である **OpenFlow** と、その利用方法、特に、**ネットワーク仮想化とプログラマビリティ**の利用方法について紹介しました。

『さて、あなたは何をしたいですか。』

■ OpenFlowの利用技術の研究開発は始まったばかりです。あなたもOpenFlowを利用した研究開発に参加してください。

■ 問い合わせ先

NEC システムプラットフォーム研究所

菅原智義 tom-sugawara@ap.jp.nec.com

Empowered by Innovation

NEC