

エンタープライズ ソリューション カンファレンス ～IT Technical Seminar V15～ (2014)

SDNフレームワークの解説と、 OpenFlowコントローラのプログラミング および実演

2014年6月20日
株式会社オープトーン
金融ソリューション事業部
菱野孝史

アジェンダ

- 第一部
 - SDN/OpenFlowとは
 - OpenFlowの仕組み
 - OpenFlowによるパケット転送の流れ
- 第二部
 - OpenFlowのアプリを作るには
 - OpenDaylight
 - FrameWorkとしてのOpenDaylight
- 第三部
 - 実演

SDNとは・・・ググってみると



ネットワークの構成、機能、性能などをソフトウェアの操作だけで動的に設定、変更できるネットワーク、あるいはそのためのコンセプトを指す



ネットワークを構成する通信機器などをソフトウェアで一括して制御することで、利用者の目的に応じてネットワークの構造、構成、設定などを柔軟に変更できる技術の総称



個々のネットワーク機器にそれぞれコンフィグレーションするのではなく、各装置を集めたネットワーク全体を一つの単位として、一括で制御するという考え方

SDNとは・・・つまり

- ソフトウェアを用いてネットワーク機器を動的に制御する技術。
- ネットワーク機器の機能を「コントロールプレーン」と「データプレーン」に分離。
- コントロールプレーンの機能を「コントロールレイヤ」として一元化することにより、複数のネットワークデバイスを集中的に制御する。

SDNとは・・・つまり

コントロールプレーン
(コントローラ)



一台のコントローラから
複数のネットワーク機器を
アプリケーションで集中制御

データプレーン
(NW機器)



SDNは何かイイのか？

- ネットワーク全体を俯瞰しながらVLANの設定ができる。ようになりそう。
- ネットワーク機器の制御を比較的自由にカスタマイズできるため、ネットワークの効率的な利用ができるようになるかも知れない。
- Oracleなど、各社からSDN対応機器が出てきそう。⇒主にネットワークの仮想化にSDN技術を用いる方向のよう。

それでは、OpenFlowとは？

- SDNにおけるネットワーク機器の制御プロトコル
- ネットワーク機器の外部にあるコントローラから、プログラムによってパケットの制御をすることができる。
- 複数のネットワーク機器を1つのコントローラから集中的に制御することができる。
- プロトコルの範囲内で、比較的自由的なパケット制御（フロー制御）ができる。



OpenFlowの仕組み

- 通信トラフィックは「フローエントリ」という形で扱う。
 - フローエントリは次の3つのフィールドを持っている。
 - ヘッダフィールド
 - OpenFlowスイッチが受信したトラフィックの要素を格納
 - トラフィックの制御に使用する
 - カウンタ
 - 発生/処理したトラフィックをヘッダフィールドの要素ごとに集計した値
 - アクション
 - トラフィックに対する処理を定義する

OpenFlowの仕組み

- トラフィックの制御はヘッダフィールドとアクションで実施する

OpenFlowの仕組み

- ヘッダフィールドの要素

フィールド	意味
Ingress Port	トラフィックの入力ポート番号
Ethernet source address	トラフィックの送信元MACアドレス
Ethernet destination address	トラフィックの宛先MACアドレス
Ethernet type	Ethernetの種類
VLAN id	VLAN ID
VLAN priority	VLAN PCP(Priority Code Point)
IP source address	送信元IPアドレス
IP destination address	宛先IPアドレス
IP protocol	トランスポート層のプロトコル種別
IP ToS bits	ToS(Type of Service) (優先度)
Transport source port / ICMP Type	送信元ポート番号
Transport destination port / ICMP Code	宛先ポート番号

OpenFlowの仕組み

- アクション

フィールド	意味
Forward	パケットを物理ポートおよび仮想ポートへ転送する
Enqueue	ポートに設定されたqueueを通してパケットを転送する queueの設定によってQoSを実現できる
Drop	パケットを転送せずに破棄する
Modify-Field	パケットの特定のフィールドを書き換える

OpenFlowのプロトコル

- コントローラとスイッチ間でやり取りするメッセージ

コントローラ⇒スイッチ		スイッチ⇒コントローラ	
メッセージ	内容	メッセージ	内容
Modify-State	スイッチのフローテーブルにフローの追加削除修正を行う ※フローにはヘッダフィールドのマッチングルールとアクションが格納される	Packet-in	スイッチがパケットを受信した際にコントローラへ通知するメッセージ コントローラは、受け取ったメッセージを解析して処理を決定する
Send-Packet	指定されたポートからパケットを送出するようにスイッチに指示を出す	Port-status	Read-Stateに対して、統計情報を返却する
Read-State	フローテーブルやポートなどの統計情報を取得する		

OpenFlowでのパケット転送の流れ

① ヘッダフィールドの要素ごとにマッチングルールを規定

マッチングルール1

フィールド	設定値
Ingress Port	1
Ethernet source address	00:11:11:11:11:11
Ethernet destination address	00:22:22:22:22:22
VLAN id	1

マッチングルール2

フィールド	設定値
Ingress Port	2
Ethernet source address	00:33:11:11:11:11
Ethernet destination address	00:33:22:22:22:22
VLAN id	2

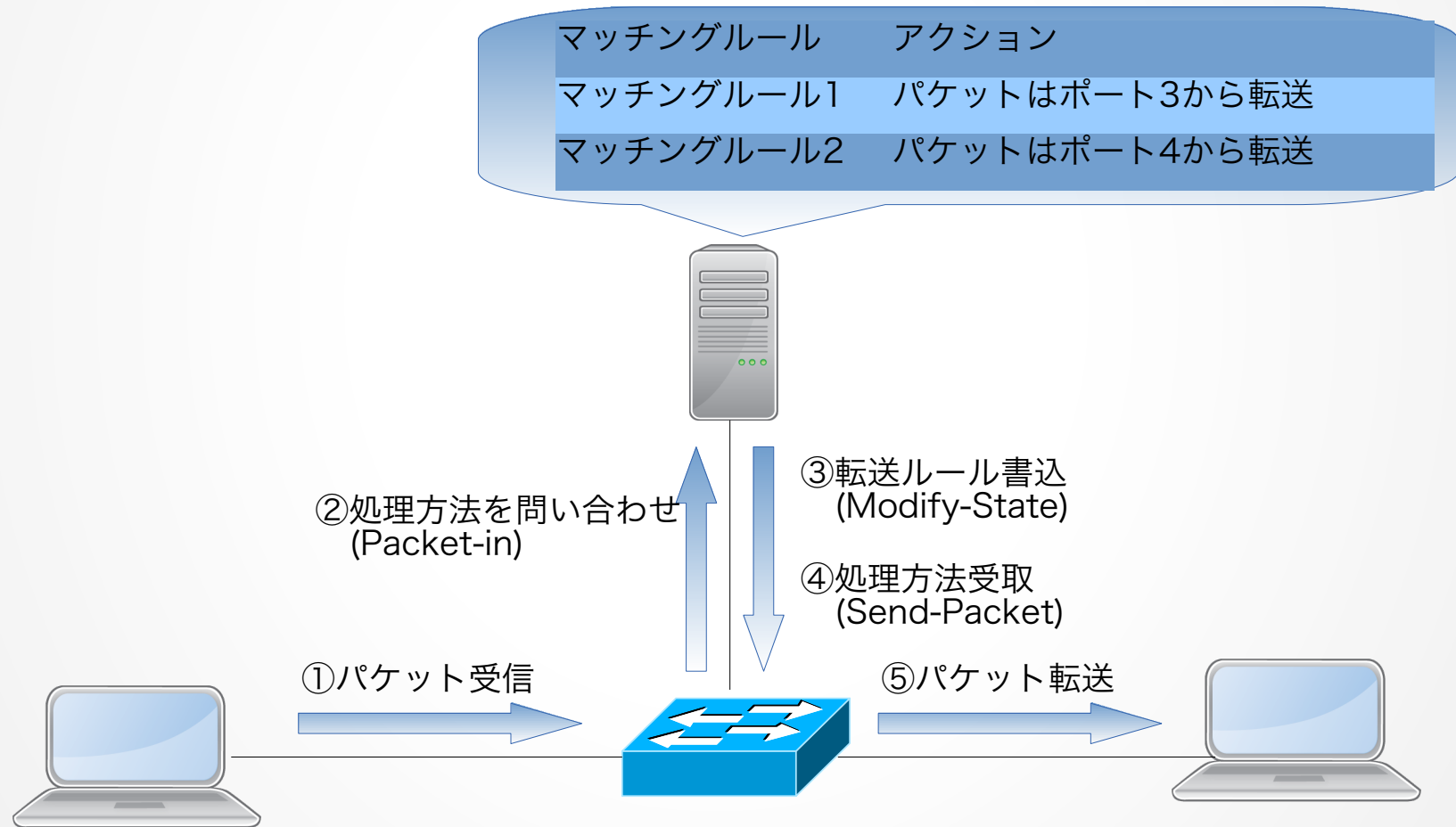
OpenFlowでのパケット転送の流れ

②マッチングルールにアクションを規定

マッチングルール	アクション
マッチングルール1	パケットはポート3から転送
マッチングルール2	パケットはポート4から転送

OpenFlowでのパケット転送の流れ

•スイッチの動作



SDN/OpenFlowのアプリを作るには

- いくつかOpenFlowに対応したFrameWorkが出てきています。

- NOX(C++)

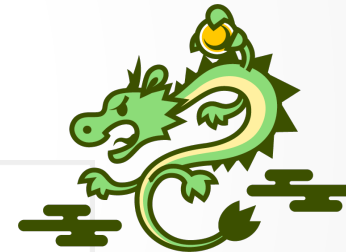
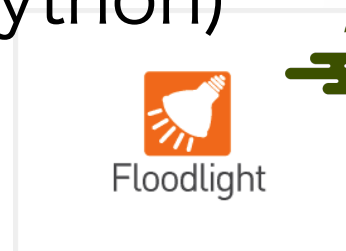


- POX(Python)



- Ryu SDN FrameWork(Python)

- Floodlight(Java)



- Trema(Ruby)

- OpenDaylight(Java)



OpenDaylight

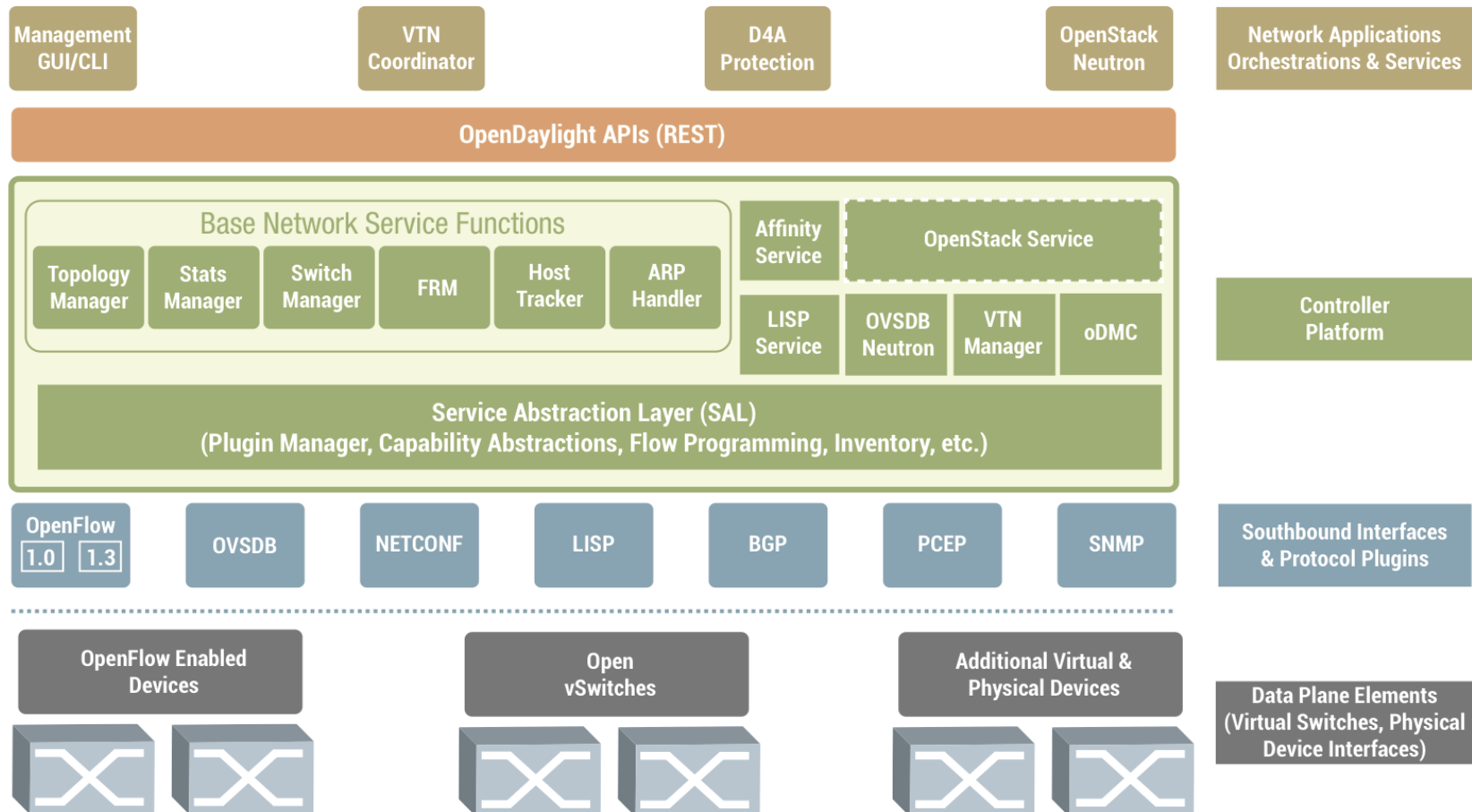
- Linux Foundationの下でのオープンソースソフトウェアプロジェクト。
- SDNフレームワークであると同時に、SDN管理機能を持っている。
- OSGiを用いて開発されたSDNコントローラの管理も可能。



OpenDaylightとは



VTN: Virtual Tenant Network
oDMC: Open Dove Management Console
D4A: Defense4All Protection
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase Protocol
BGP: Border Gateway Protocol
PCEP: Path Computation Element Communication Protocol
SNMP: Simple Network Management Protocol
FRM: Forwarding Rules Manager
ARP: Address Resolution Protocol



OpenDaylightで何ができるのか

- OpenDaylight対応機器で構成されたネットワークの全体を管理することが可能。
- OpenDaylight自身にフローコントロールの機能が実装されており、パケットの制御をカスタマイズできる。
- OpenFlowに対応しており、OpenFlowのFrameWorkとして利用可能。

FrameWorkとしての OpenDaylight

- Base Network Service Function
 - Java Class

バンドル	API	説明
arphandler	IHostFinder	ARPを処理してホストの場所を学習する
hosttracker	IlftoHost	SDN上でホストの相対的な場所を追跡する
switchmanager	ISwitchManager	コントローラ内のすべてのスイッチのインベントリを保持する
topologymanager	ITopologyManager	ネットワーク全体のトポロジを保持する
userManager	IUserManager	ユーザ管理を担う
statisticsmanager	IStatisticsManager	IReadServiceを利用して統計情報を収集する

FrameWorkとしての OpenDaylight

- Service Abstraction Layer
 - Java Class

バンドル	API	説明
sal	IReadService	スイッチのフロー/ポート/キューのビューを取得する
sal	ITopologyService	トポロジメソッド
sal	IFlowProgrammerService	スイッチに対してフローエントリの追加更新削除を行う
sal	IDataPacketService	パケットデコード等、パケット操作を行う
sal	IListenDataPacket	パケット処理のためのクラス このクラスを継承してコントローラクラスを作成する
web	IStatisticsManager	システムにインストールされているバンドルをUIから管理する

FrameWorkとしての OpenDaylight

- Service Abstraction Layer
 - JavaのClass

バンドル	Class	説明
sal	Action	OpenFlowのアクション
sal	Match	OpenFlowのマッチ
sal	IFlowProgrammerService	スイッチに対してフローエントリの追加更新削除を行う
sal	IDataPacketService	パケット操作のためのサービス OpenFlowのForwardアクションを発行する
web	IStatisticsManager	システムにインストールされているバンドルをUIから管理する

開発・テストツール

- OpenDaylight
- OpenWrt
- サンプルコントローラ
https://github.com/sdnhub/SDNHub_Opendaylight_Tutorial

OpenWrt

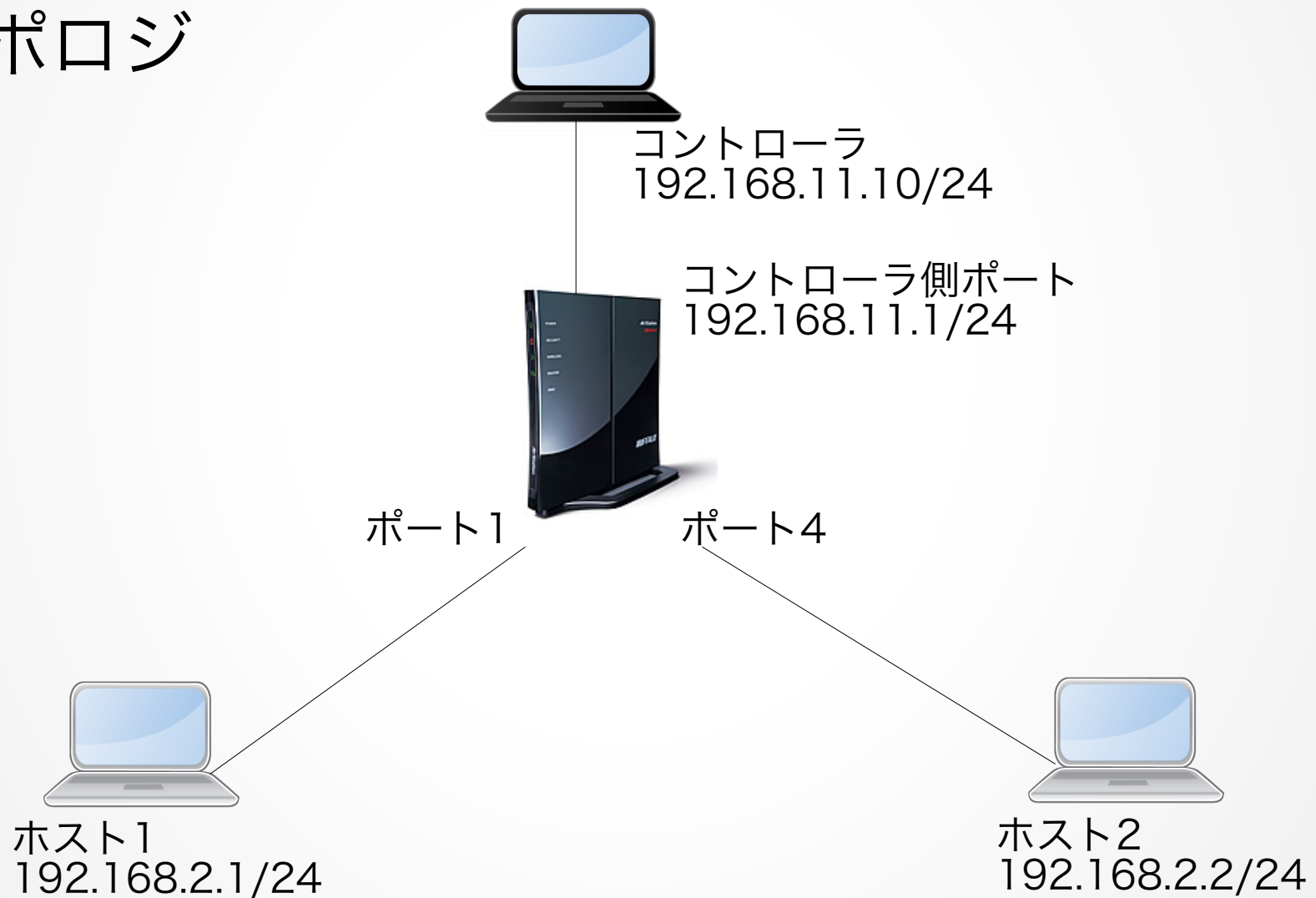
- 組み込みデバイス向けのLinuxディストリビューション
- 一部の安価な無線LANルータを動かすことができる。
- OpenWrt向けのOpenFlowがある。
以下のサイトで入手可能。
<http://openflow.inthefox.info>

実演

- 環境
 - OpenFlowコントローラ
 - CentOS 6.5 (Virtual Box 4.3上で動作)
 - OpenDaylight Hydrogen
 - テストルータ
 - BUFFALO WHR-G301N
 - OpenWrt
 - OpenFlow 1.0

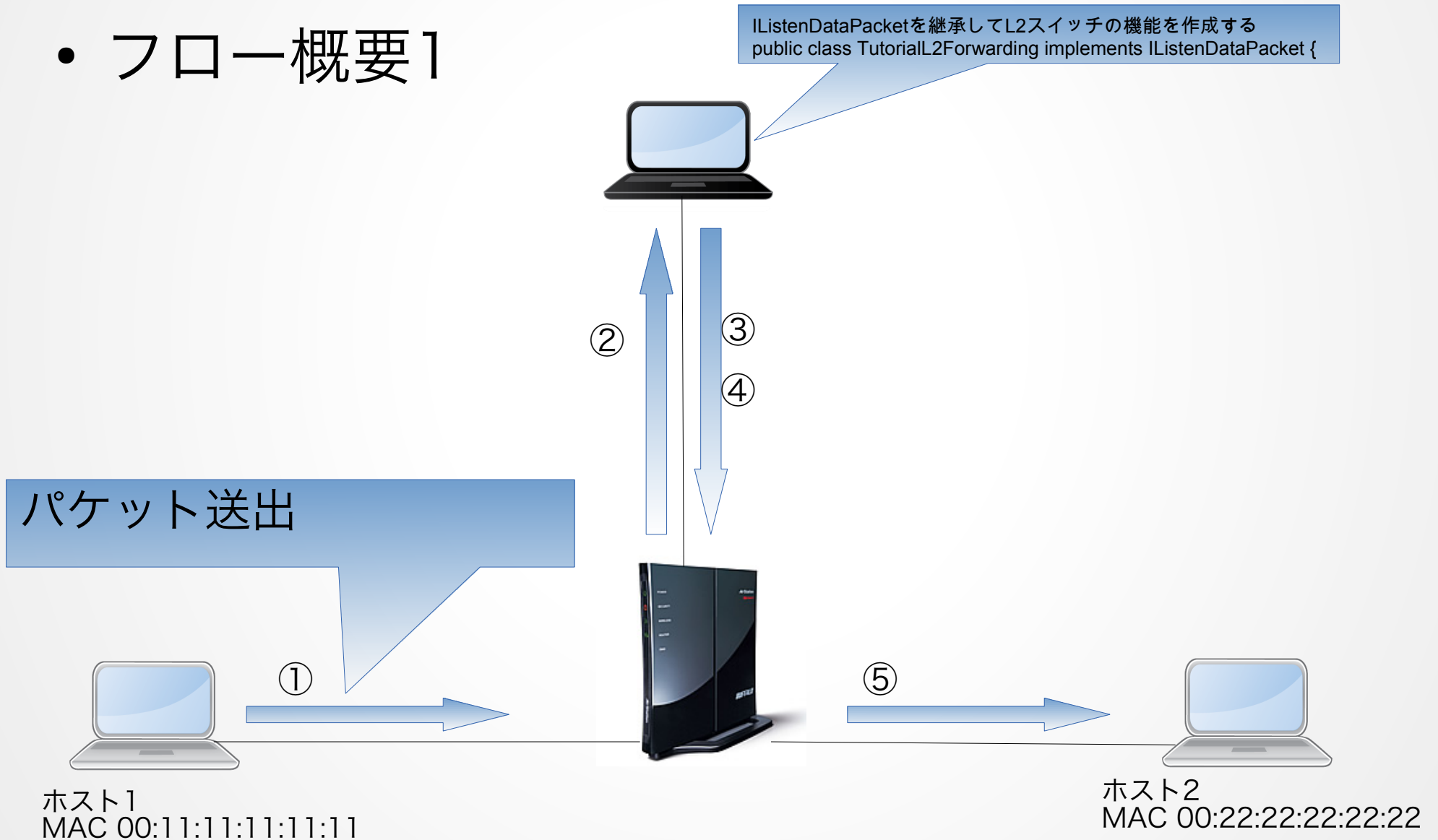
実演

- トポロジ



実演

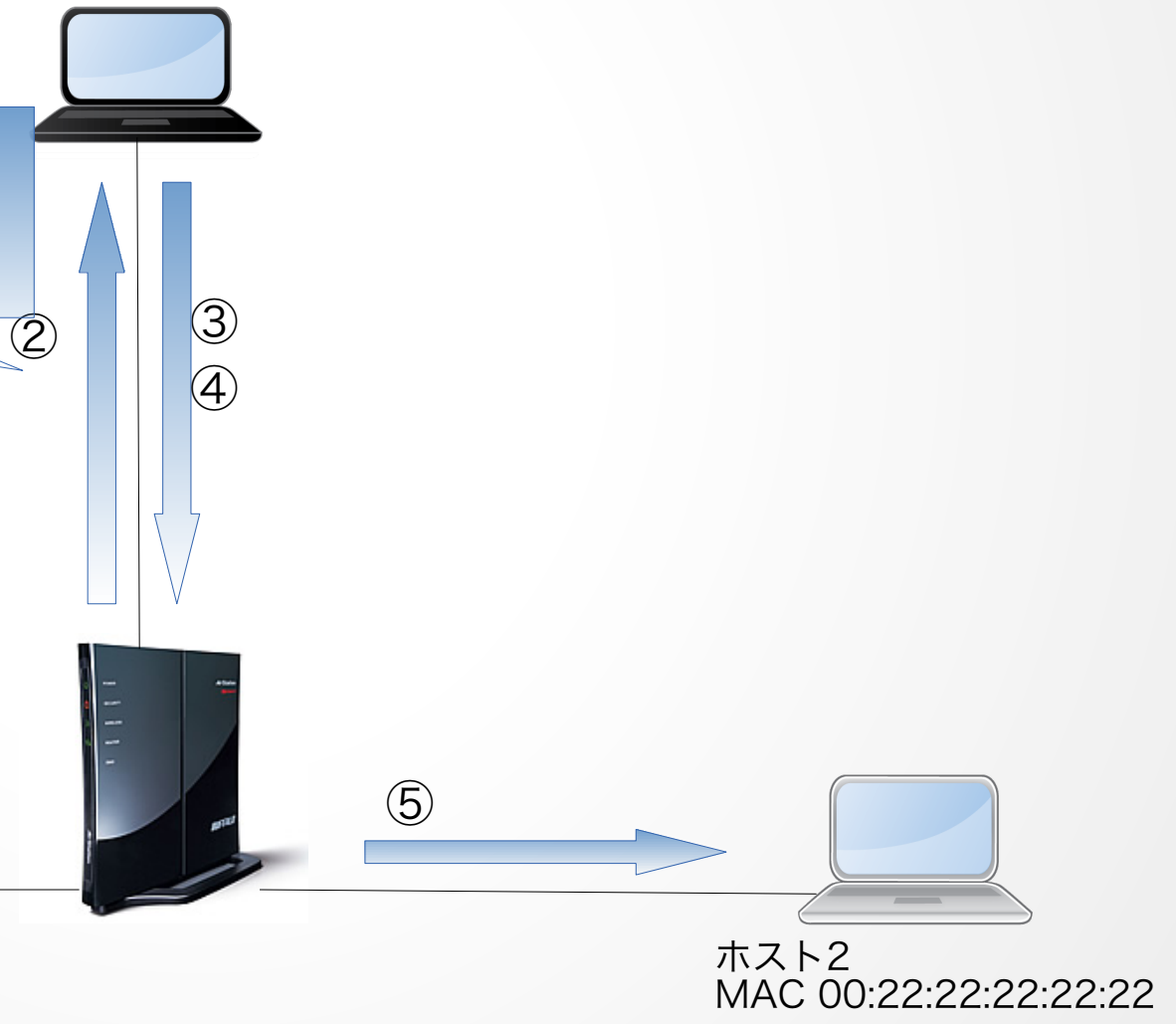
- フロー概要1



実演

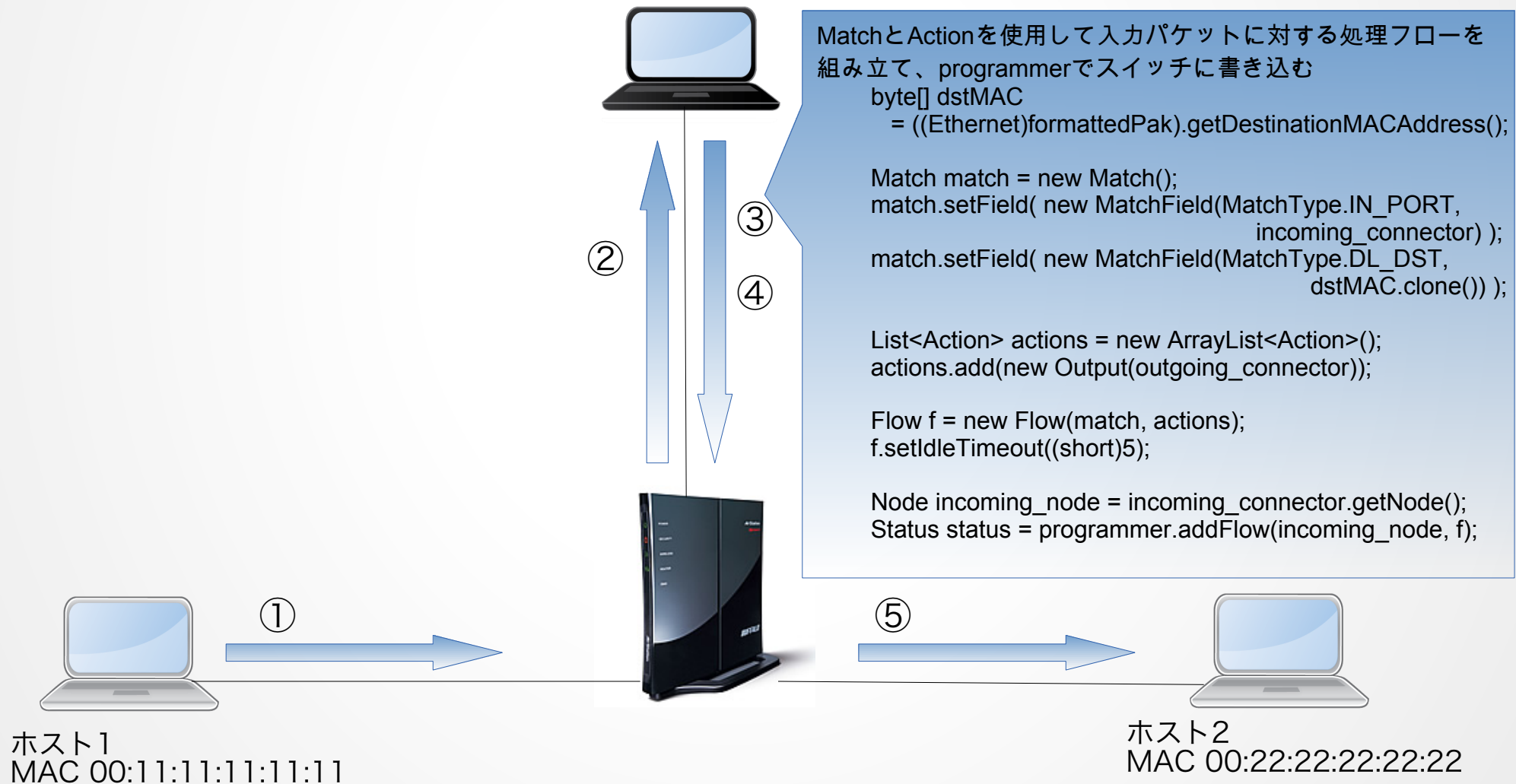
• フロー概要2

Packt-inでコントローラの処理開始
IListenDataPacketの
receiveDataPacketメソッドが呼び出される。
public PacketResult
receiveDataPacket(RawPacket inPkt) {



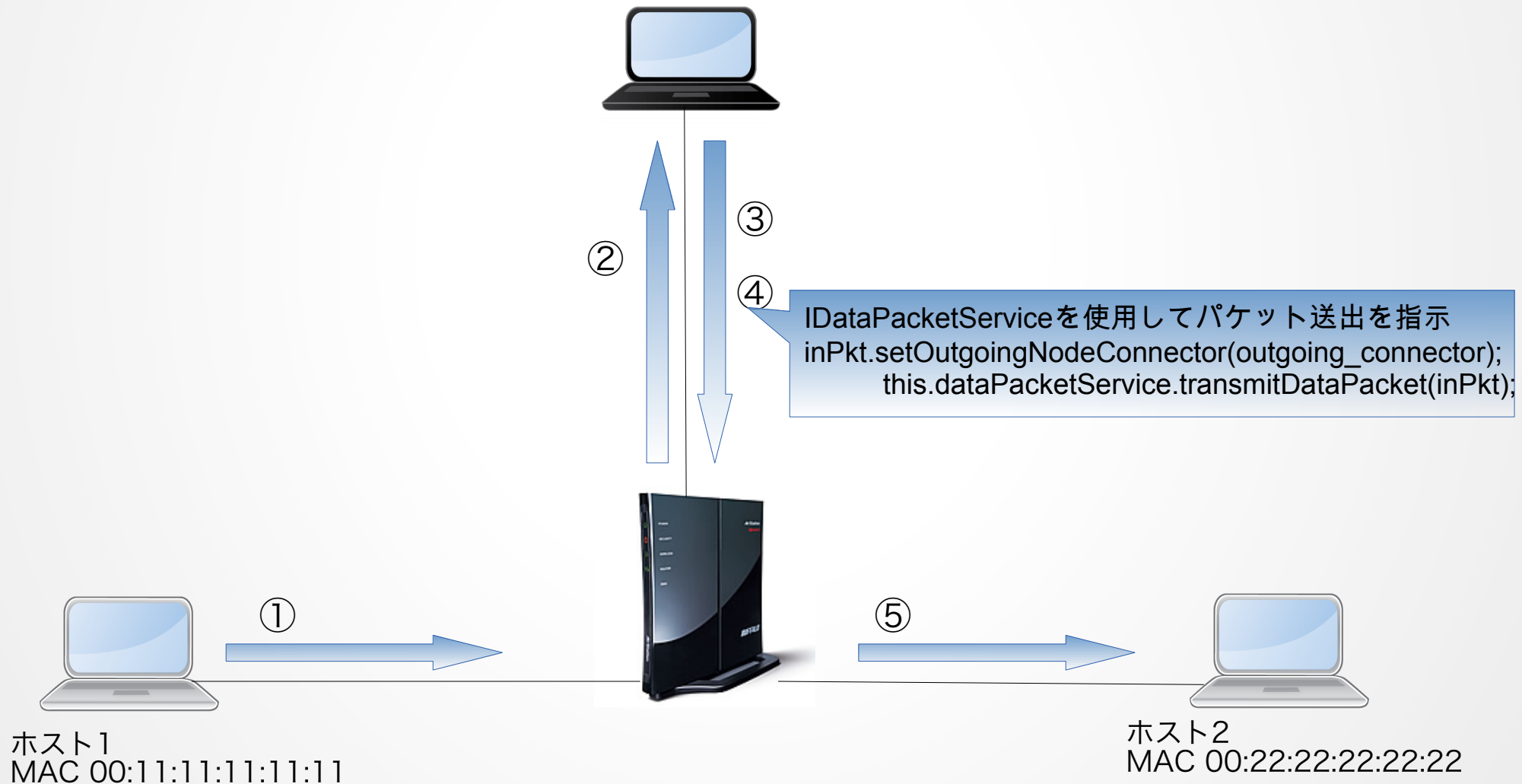
実演

• フロー概要3



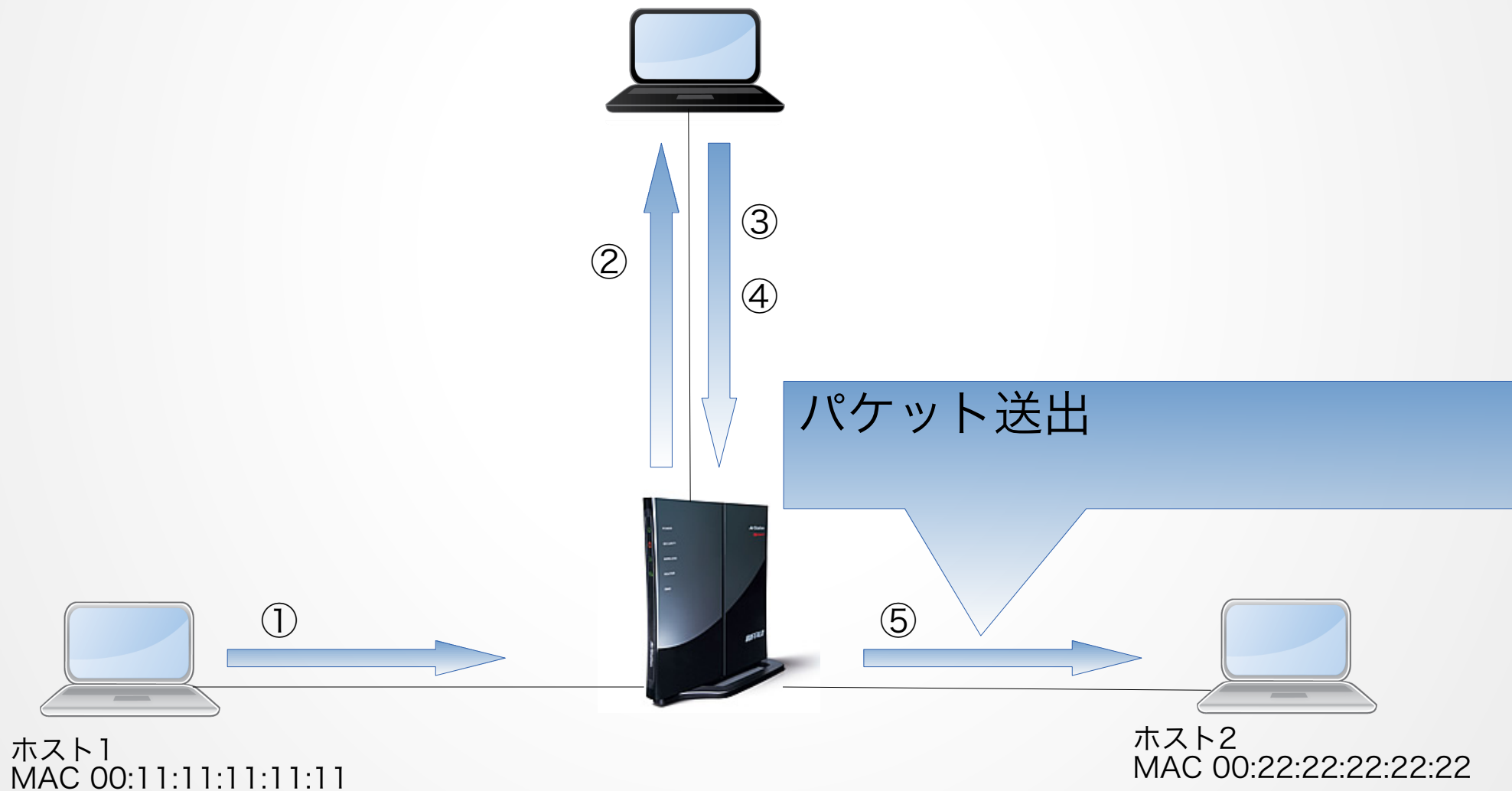
実演

- フロー概要4



実演

- フロー概要5



終わりに



本セミナーで実演した内容など、
SDN/OpenFlow/OpenDaylightに関する記事は、随時弊社ブログに掲載していきます。
ご興味のある方は是非ご覧ください。
<http://labs.opentone.co.jp>

エンタープライズ ソリューション カンファレンス ～IT Technical Seminar V15～ (2014)

ご静聴、ありがとうございました。