OpenFlow勉強会 (20min Talk, 5min Q&A)
http://atnd.org/events/19554

OpenFlow標準化動向とProtocolの基本動作概要

2011年9月20日

日本電気株式会社
岩田　淳
E-mail: a-iwata@ah.jp.nec.com

Outline

- OpenFlowの標準化動向
- OpenFlowの基礎
- OpenFlowの動作概要
- OpenFlow Controller Platform: Trema （NEC’s Open Source Platform）
- クラウドデータセンター向けProgrammableFlowソリューション

OpenFlowの標準化動向

OpenFlow標準化動向(１)

OpenFlowコンソーシアムにて標準スペックを策定（2007以後）

- http://www.openflow.org/
- Stanford大学のProf. Nick McKeownが提唱
- OpenFlowプロトコル標準スペック「OpenFlow Switch Specification」を策定しており、Ver 1.0 仕様化 @ 2010.1, Ver1.1仕様化 @2011.2
- メンバー:
  - NEC, HP, Juniper, Cisco, Arista, Nicira, Big Switch Network, Ericsson, Broadcom, Dell Google, DT Lab, NTT DoCoMo lab, FT/Orange Lab　他
- NECは創立時点から加盟。OpenFlowスペック策定、トライアル支援。最新スペック対応スイッチ試作機を他社に先駆けて順次リリース(0.8.9 @ 2008.11, 1.0 @ 2010.4)。Campus/Internet2/NICT JGN2plus トライアル成功。
- NECはOpenFlow 1.0準拠のスイッチ、コントローラ両方の製品化を世界初実現し、NEC独自のProgrammableFlow機能を追加し、データセンタソリューション化。
2011年4月：世界に先駆け商用製品を販売開始。

ProgrammableFlowスイッチ: PF5240

ProgrammableFlow コントローラ: PF6800

OpenFlow標準化動向(２)

- Open Networking Foundation (ONF) の設立: 2011/3/21 PST
  - http://www.opennetworkingfoundation.org/
  - Non Profit Organization,　Dan Pitt がExecutive Directorへ就任。
- 目的:
  - Software-Defined Networking (SDN) (OpenFlow) のpromotion と標準化
  - OpenFlow Switch Consortium からOfficial標準化活動部分のみが分離。
- 参加会社 (2011/6/20時点)
  - ボードメンバー: 5 社
    - キャリア: Deutsche Telekom, Verizon
    - クラウドプロバイダ: Google, Facebook, Yahoo, Microsoft (#1)
  - メンバー: 35 社
    - キャリア: NTT, Comcast
    - チップベンダ: Broadcom, Marvell, Intel, Netronome,
    - スイッチ/コントローラベンダ: NEC, HP, Cisco, Juniper Networks, IBM, Dell, Brocade, Force10 Networks, Extreme Networks, Fujitsu, Mellanox Technologies, Huawei, Netgear, Ericsson, Ciena, Nokia Siemens, Meta Switch Networks, Big Switch Network, Nicira Networks, Midokura, IP Infusion, Plexxi, Pronto Systems, Vello Systems
    - 仮想化ソフトウエアベンダ: VMWare, Citrix, Microsoft (#2)
    - アプライアンスベンダ: Riverbed Technology
    - 測定機: IXIA,
    - その他: CompTIA
- 2011/6/21にBay Areaにて第一回標準化会合開催

Empowered by Innovation NEC

OpenFlow標準化動向(３)

ONF団体標準化Policy

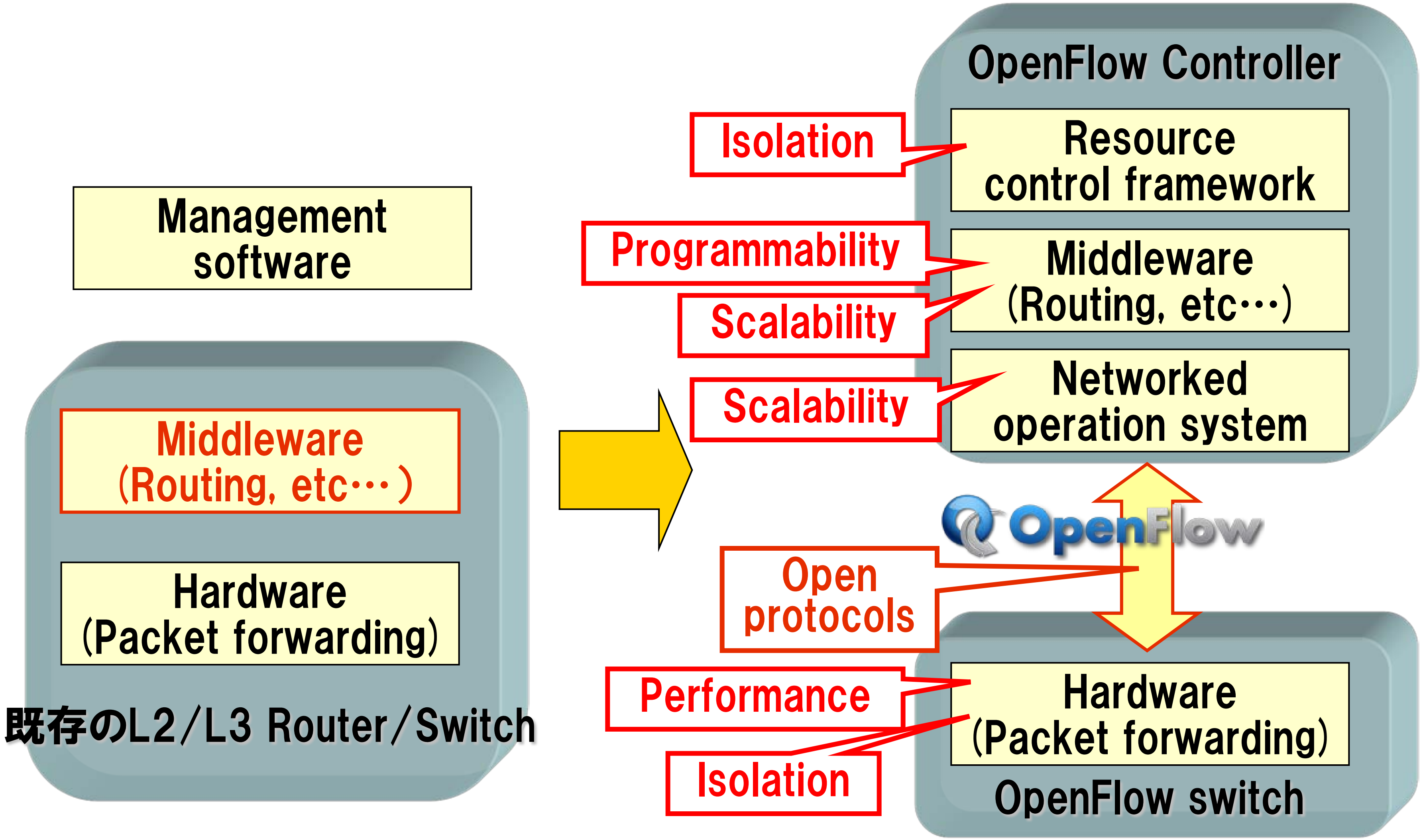
- グループ会社の場合代表1社が契約締結 (e.g. NTT様など)
- 標準化での議論内容は、参加会社以外には開示が不可能
- Working Group Itemでの議論テーマについてはBoard Memberでの承認後、各技術部会に分かれて議論。地理的な問題もあり、基本的には Mail & Voice Conferenceベース。

標準化Topic (詳細についてはONFに加盟して入手お願いします)

- IPv6対応
- メッセージ拡張性対応
- 各種拡張 Configuration対応 (QoS, Controller redundancy, etc)
- Topology Discovery方式
- Controller Discovery方式
- Conformance Test & Interoperability

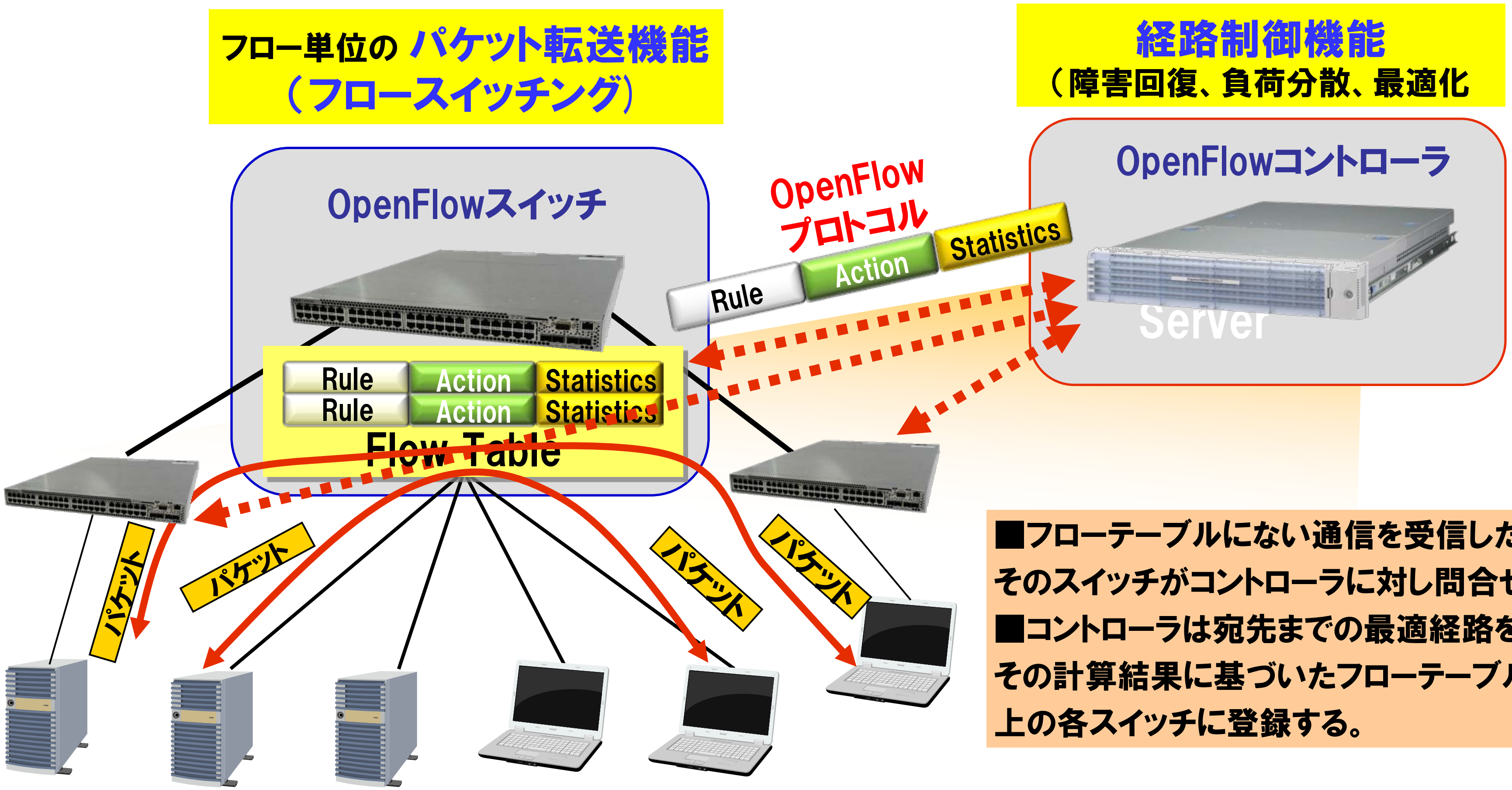
OpenFlowの基礎

OpenFlow基本ノードアーキテクチャ



OpenFlow技術による集中制御のしくみ

- パケット転送機能と経路制御機能をフロー制御プロトコルにより分離
- 通信トラフィックをフロー単位に制御することで、
  経路制御の高度化、ネットワーク仮想化、可視化等を実現



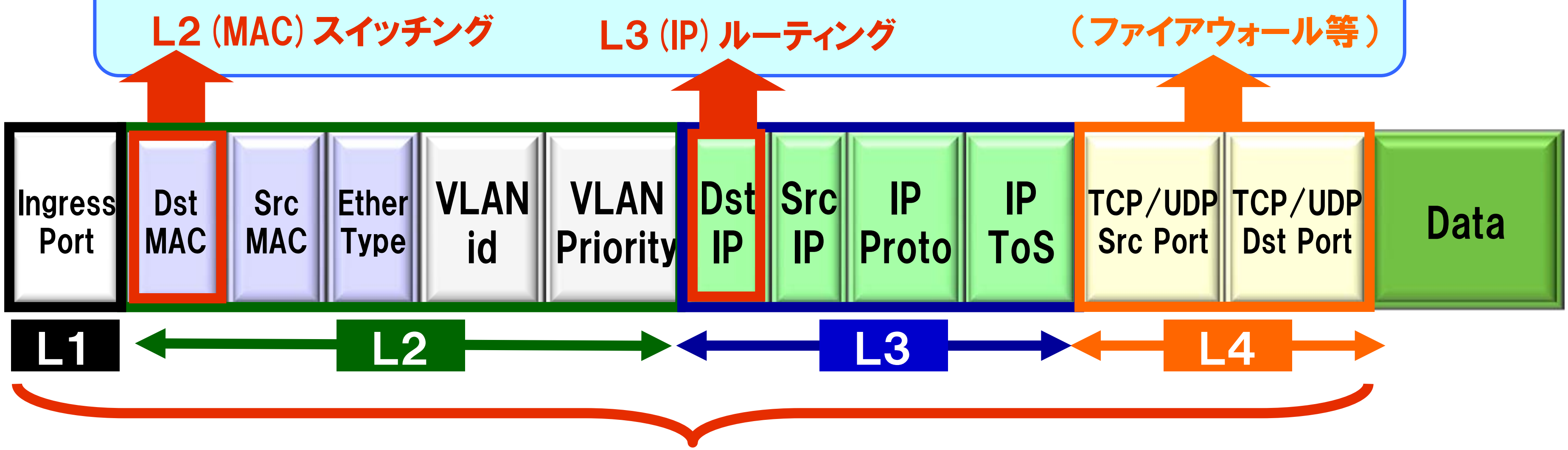
■フローテーブルにない通信を受信した場合は、そのスイッチがコントローラに対し問合せを実施。
■コントローラは宛先までの最適経路を計算し、その計算結果に基づいたフローテーブルを経路上の各スイッチに登録する。

Empowered by Innovation NEC

フロースイッチングとは？

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います。

【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します



【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です。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んでいます。

フローテーブルの定義と高度な転送機能

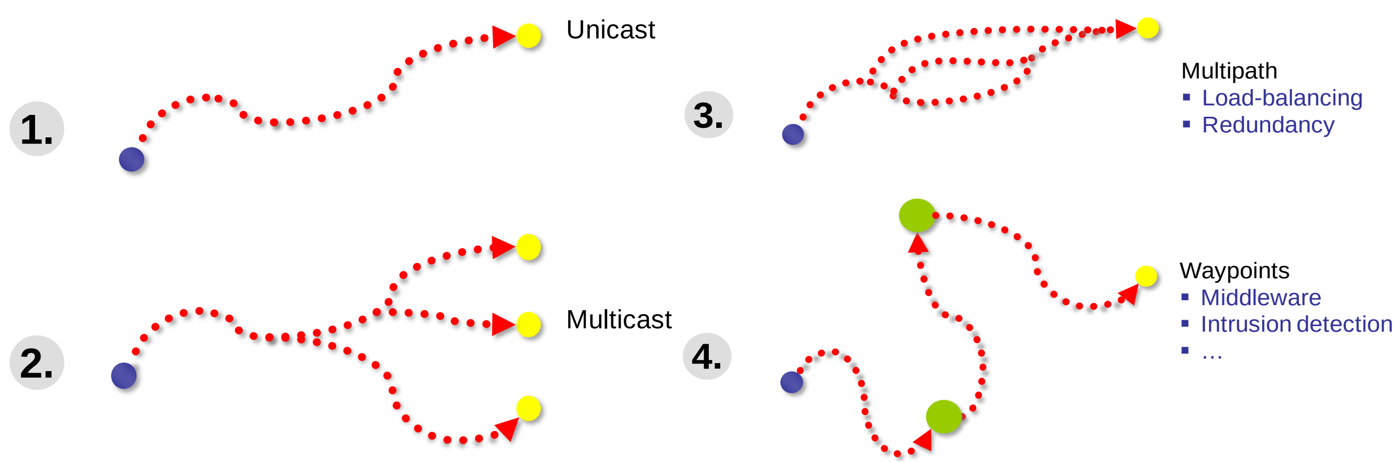
Flow 1.	Rule (exact & wildcard)	Action	Statistics
Flow N.	Rule (exact & wildcard)	Default Action	Statistics

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

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,

フロー統計情報
（例）
パケット数、バイト数、セッション継続時間

Action例

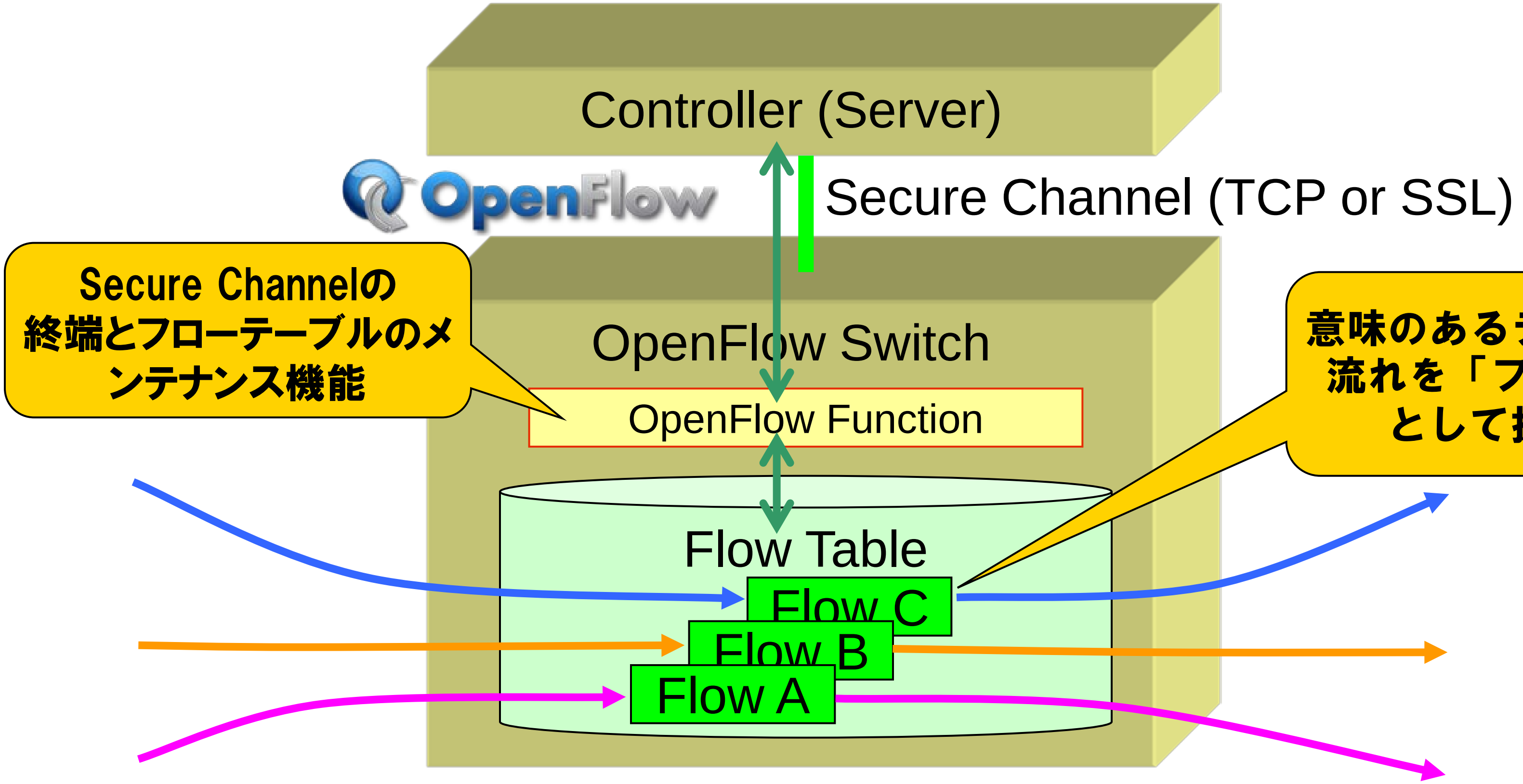


OpenFlow動作概要

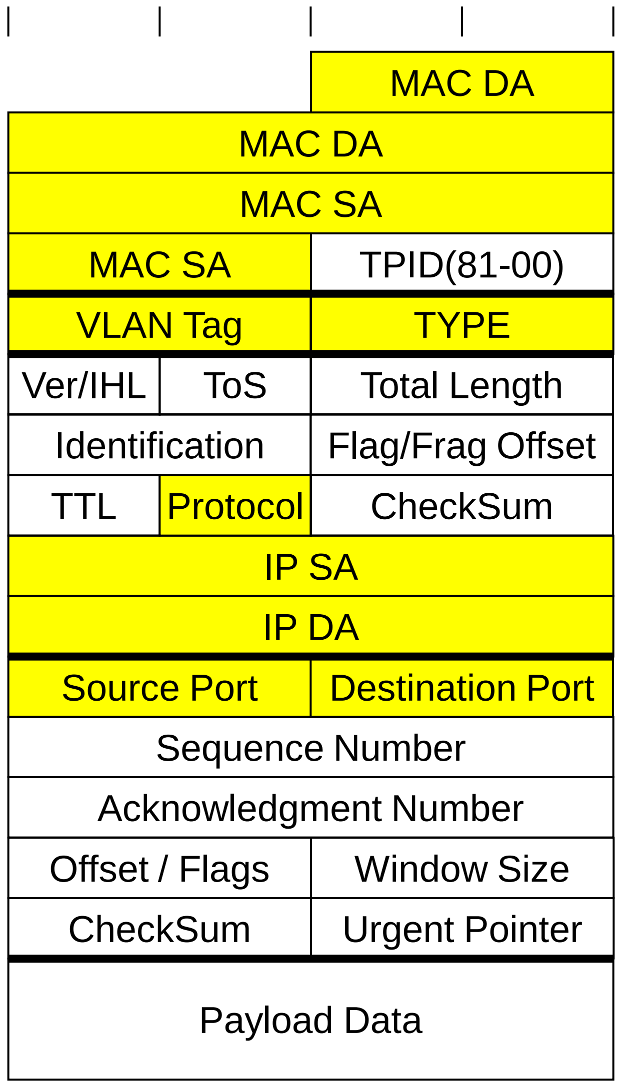
OpenFlowの基本概念

- **コントローラとスイッチは**Secure Channel**で接続され、コントローラは**Secure Channel**を通じてフローテーブルをプログラムする。**

- **スイッチに入ってきたパケットはフローに従って処理・出力される。**



フローの検索キーとして使用可能なフィールド



レイヤ	フィールド
Physical	Input Port No
Ethernet	MAC DA, MAC SA, VLAN Tag※, Type
IPv4	IP SA, IP DA, Protocol
ICMP	ICMP Type, ICMP Code
TCP/UDP	Source Port, Destination Port

※ VLAN Tag内で検索キーとして使えるのはVLAN Priority BitとVLAN IDの2つ。

検索キーのWildcardについて

- 検索キーの各フィールドはWildcardとして指定可能。
  即ち、Wildcardと指定されたフィールドはany扱いになる。
  （例：IPDAだけをキーとしてもつフローなどを定義できる）

- IPDAとIPSAはWildcardの指定時にPrefix長も指定可能。
  （例：IPSA=10.20.x.x/16の様な条件を定義できる）

- Wildcardの指定がないものExact Matchフローと呼ぶ。

- Wildcardの指定があるものをWildcard Matchフローと呼ぶ。

フローの種類とプライオリティの関係

優先度高

- Exact Match Flow A
- Exact Match Flow B
- Exact Match Flow C

Exact Matchのフローは常に最高優先度

- Wildcard Flow D (Priority = 65535)

←最大Priorityは65535

- Wildcard Flow E (Priority = 100)
- Wildcard Flow F (Priority = 100)

統一Priority内の優先度は規定しない（実装依存）

- Wildcard Flow G (Priority = 0)

←最小Priorityは0

優先度低

フローに設定可能なアクション

※赤色でハイライトしているのは、定常状態でよく使われるメッセージです。

アクション名	アクションの内容
OUTPUT	指定ポートに出力する。
SET_VLAN_VID	指定VLAN IDでVLAN Tagを追加・更新する。
SET_VLAN_PCP	指定VLAN PriorityでVLAN Tagを追加・更新する。
STRIP_VLAN	IEEE802.1q VLAN Tagを外す。
SET_DL_SRC	MAC SAを更新する。
SET_DL_DST	MAC DAを更新する。
SET_NW_SRC	IP SAを更新する。
SET_NW_DST	IP DAを更新する。
SET_NW_TOS	IPヘッダにあるToSフィールド上のDSCP bitを更新する。
SET_TP_SRC	TCP/UDP Source Portを更新する。
SET_TP_DST	TCP/UDP Destination Portを更新する。
VENDOR	ベンダ定義アクション。

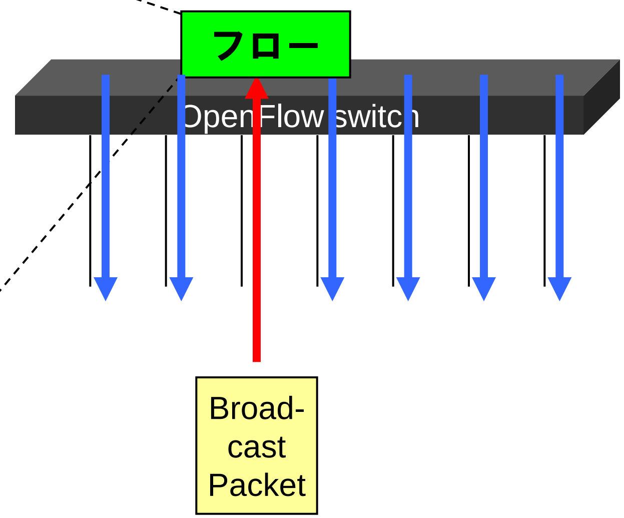
入出力ポートとして利用可能な仮想ポート定義

仮想ポート名	意味
IN_PORT	パケットを受信したポートを意味する。
NORMAL	パケットをOpenFlowの制御下から既存のプロトコル処理（BridgeやIP Forwarding）に受け渡す。
FLOOD	パケットを受信したポート以外のForwarding状態の全てのポート。
ALL	パケットを受信したポート以外の全てのポート。
CONTROLLER	Secure Channelを経由してのコントローラ宛て。
LOCAL	スイッチ自身のネットワークスタック行き。（＝ソケット行き）

フローの設定例（1）

➢どんなポートから受け取ったBroadcastパケットも受信ポート以外に送信

検索Key	Input Port = *	MAC SA	
	MAC SA = *		
	MAC DA = FF:FF:FF:FF:FF:FF		
	MAC DA	VLAN ID = *	
	TYPE = *	Proto = *	0
	IP SA = *		
	IP DA = *		
	Src Port = *	Dest Port = *	
Action 1	OUTPUT = FLOOD		

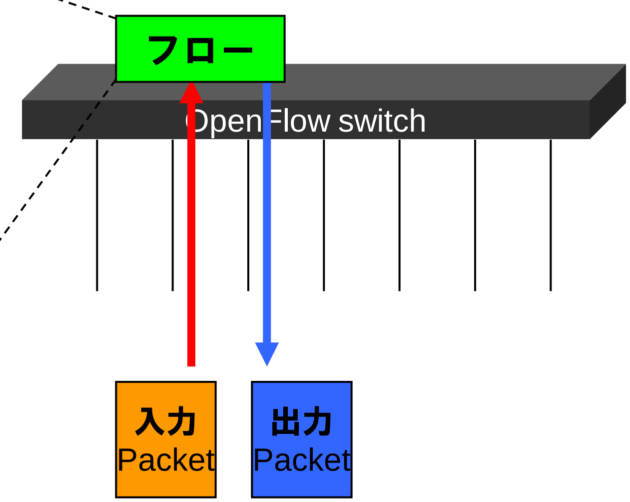


※見易さを考慮して、VLAN TagフィールドをVLAN IDとして表記。

フローの設定例(2)

➢Waypoint　Routing/Redirect処理
➢ポート2から受信した10.20.30.40→10.50.60.70行きのHTTP通信をMAC DAを次装置のものに書き換えてポート3に送信

検索Key	Input Port = 2	MAC SA	
	MAC SA = 00:00:4c:11:11:11		
	MAC DA = 00:00:4c:22:22:22		
	MAC DA	VLAN ID = 100	
	TYPE = 0x0800	Proto = 6	0
	IP SA = 10.20.30.40		
	IP DA = 10.50.60.70		
	Src Port = 32000	Dest Port = 80	
Action 1	SET_DL_DST = 00:00:4c:33:33:33		
Action 2	OUTPUT = 3		



※見易さを考慮して、VLAN TagフィールドをVLAN IDとして表記。

Secure Channel上で使われるメッセージ一覧

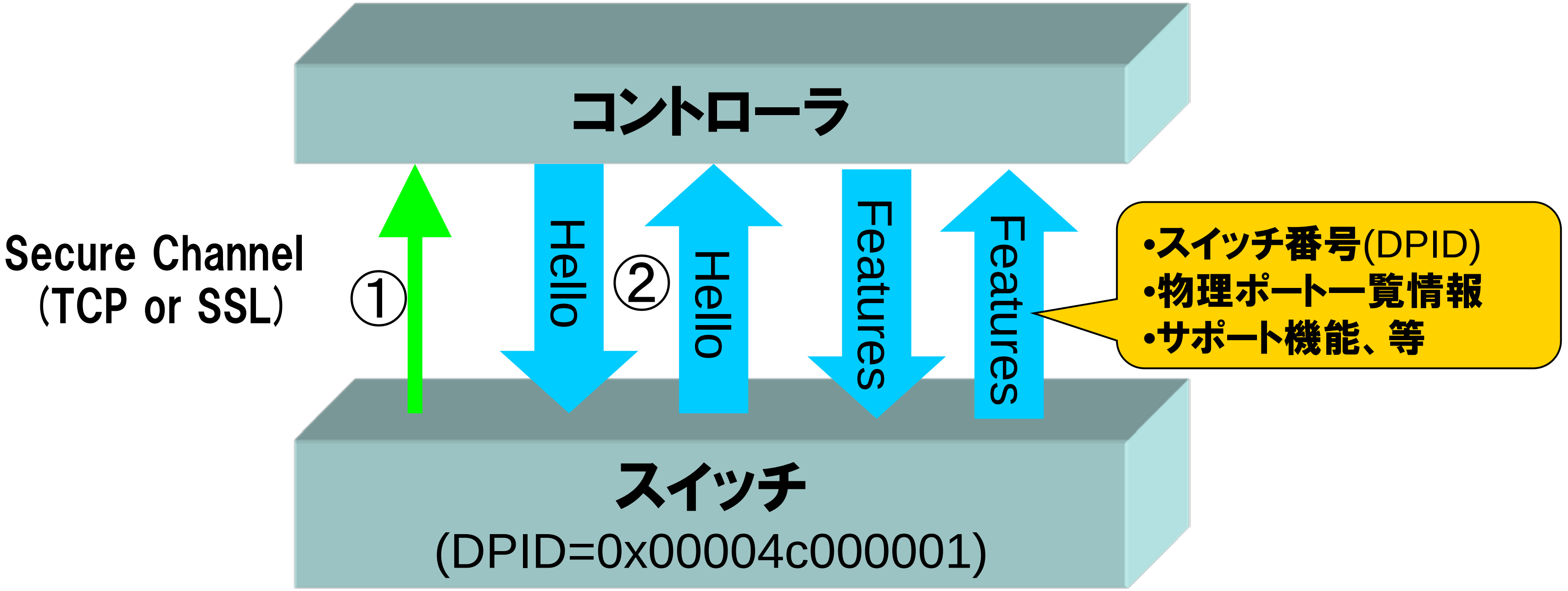
メッセージ名	方向	用途
Hello	CTRL ⇔ SW	バージョンネゴシエーションに使用する。
Error	CTRL ⇔ SW	メッセージにエラーがあった事を応答する。
Echo Request	CTRL ⇔ SW	Keep AliveのためのEchoを要求する。
Echo Reply	CTRL ⇔ SW	Keep AliveのためのEchoに応答する。
Vendor	CTRL ⇔ SW	ベンダ定義メッセージ。
Features Request	CTRL ⇒ SW	スイッチの機能・特徴の問い合わせを要求する。
Features Reply	SW ⇒ CTRL	スイッチの機能・特徴の問い合わせに応答する。
Get Configuration Request	CTRL ⇒ SW	スイッチのコンフィグレーションの取得を要求する。
Get Configuration Reply	SW ⇒ CTRL	スイッチのコンフィグレーションの取得に応答する。
Set Configuration	CTRL ⇒ SW	スイッチのコンフィグレーションを設定する。
Packet In	SW ⇒ CTRL	スイッチに入力されたパケットをコントローラに通知する。
Flow Removed	SW ⇒ CTRL	フローが満了時刻を迎えたことをコントローラに通知する。
Port Status	SW ⇒ CTRL	物理ポートの状態変化をコントローラに通知する。
Packet Out	CTRL ⇔ SW	コントローラからスイッチにパケットの出力を指示する。
Flow Mod	CTRL ⇔ SW	コントローラがスイッチにフローの登録・変更・削除を要求する。
Port Mod	CTRL ⇔ SW	物理ポートの状態変更を要求する。
Statistics Request	CTRL ⇒ SW	統計情報取得を要求する。
Statistics Reply	SW ⇒ CTRL	統計情報取得に応答する。
Barrier Request	CTRL ⇒ SW	スイッチ上でこれまで受信したメッセージ処理が完了したか確認する。
Barrier Reply	SW ⇒ CTRL	スイッチ上でこれまで受信したメッセージ処理が完了したことを通知する。

Secure Channelの接続、バージョン／機能ネゴシエーション

①Secure Channelの接続
- OpenFlowとしてSecure Channelの接続を開始する側の規定は無 いが、スイッチ側が接続を開始する実装が多い。
- SSLの場合は、スイッチの証明書の確認も行われる。

②バージョンネゴシエーション
- Helloメッセージを出し合い、お互いのサポートバージョンを確認しあう。

③機能(Features)ネゴシエーション
- スイッチの機能確認を行う。



Empowered by Innovation NEC

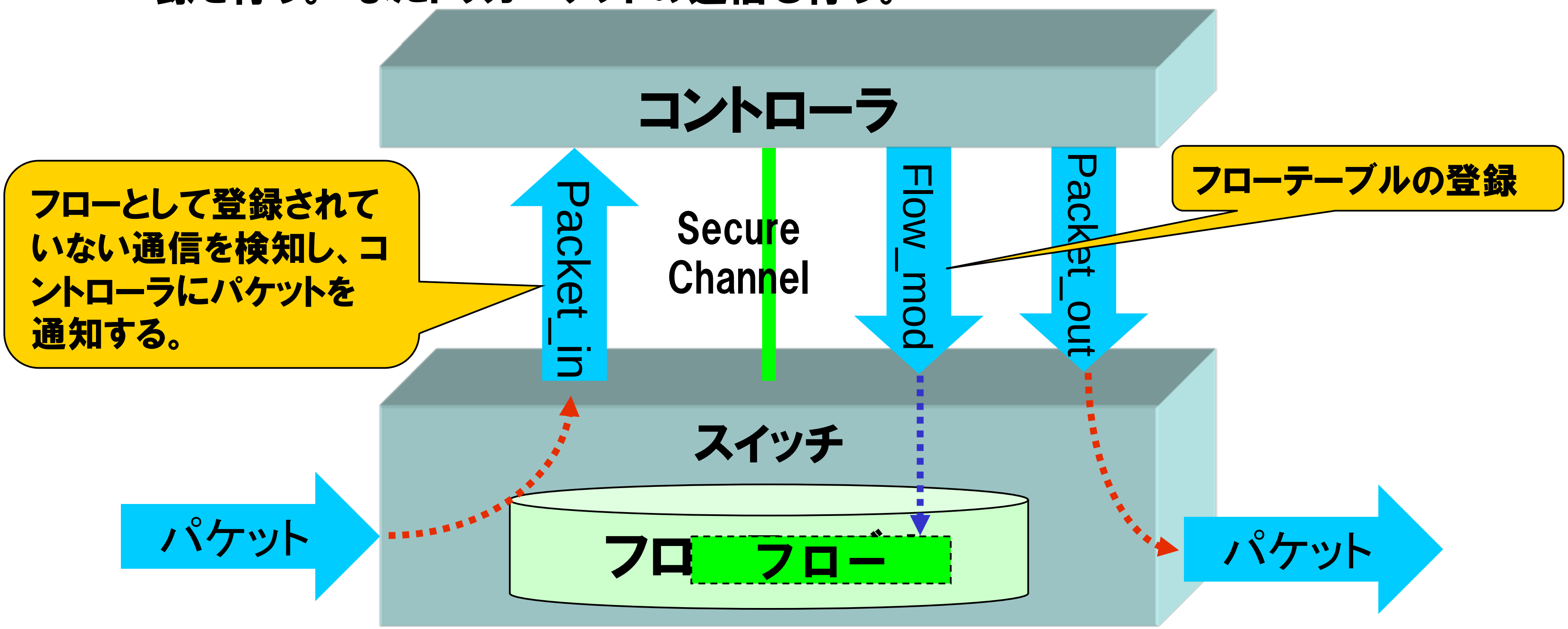
新規フロー通信の検出、フローの登録とパケットの送信

④新規フロー通信の検出

➢スイッチはフローとして登録されていない通信を検知すると、コントローラにパケットを通知する。パケット全体を送るか先頭一部を送るか選択可能。

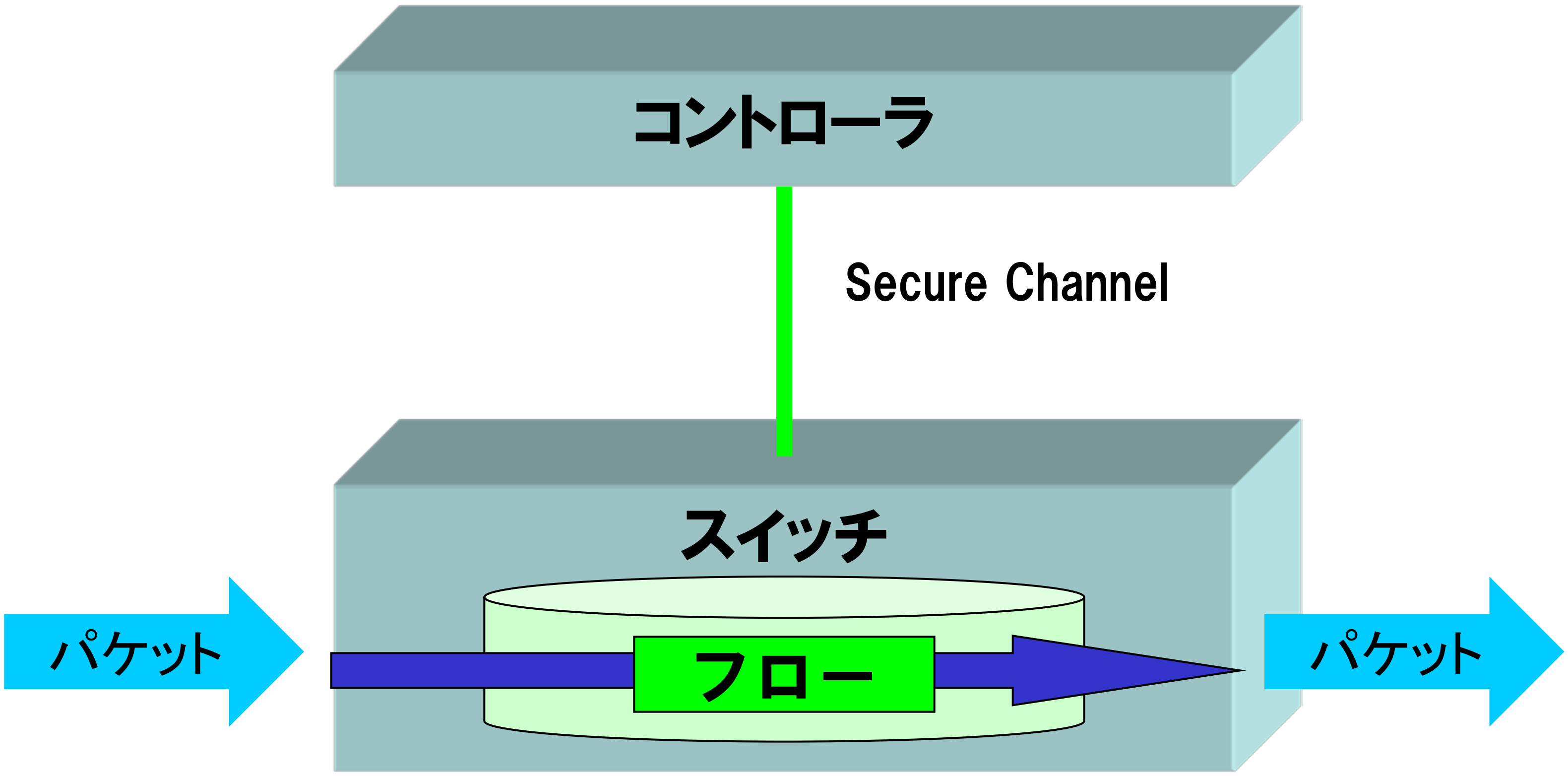
⑤フローの登録とパケットの送信

➢新規通信の発生などのトリガを利用し、コントローラはスイッチにフローの登録を行う。 またトリガパケットの送信も行う。



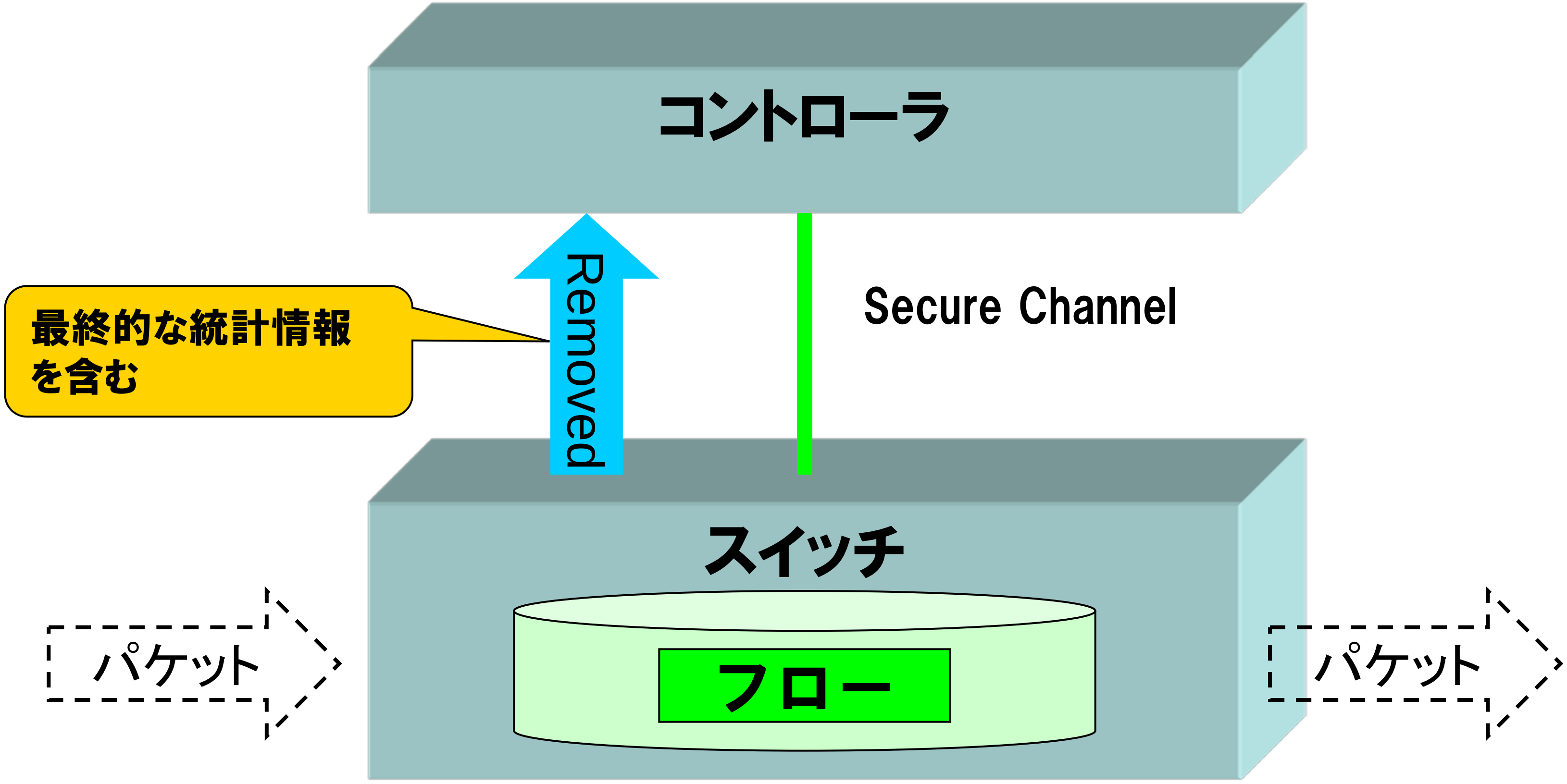
パケット転送(通信の継続)

- フローにはAging時間を設定でき、時間内であれば登録済みフローのパケットはアクションに従いスイッチで転送処理される。



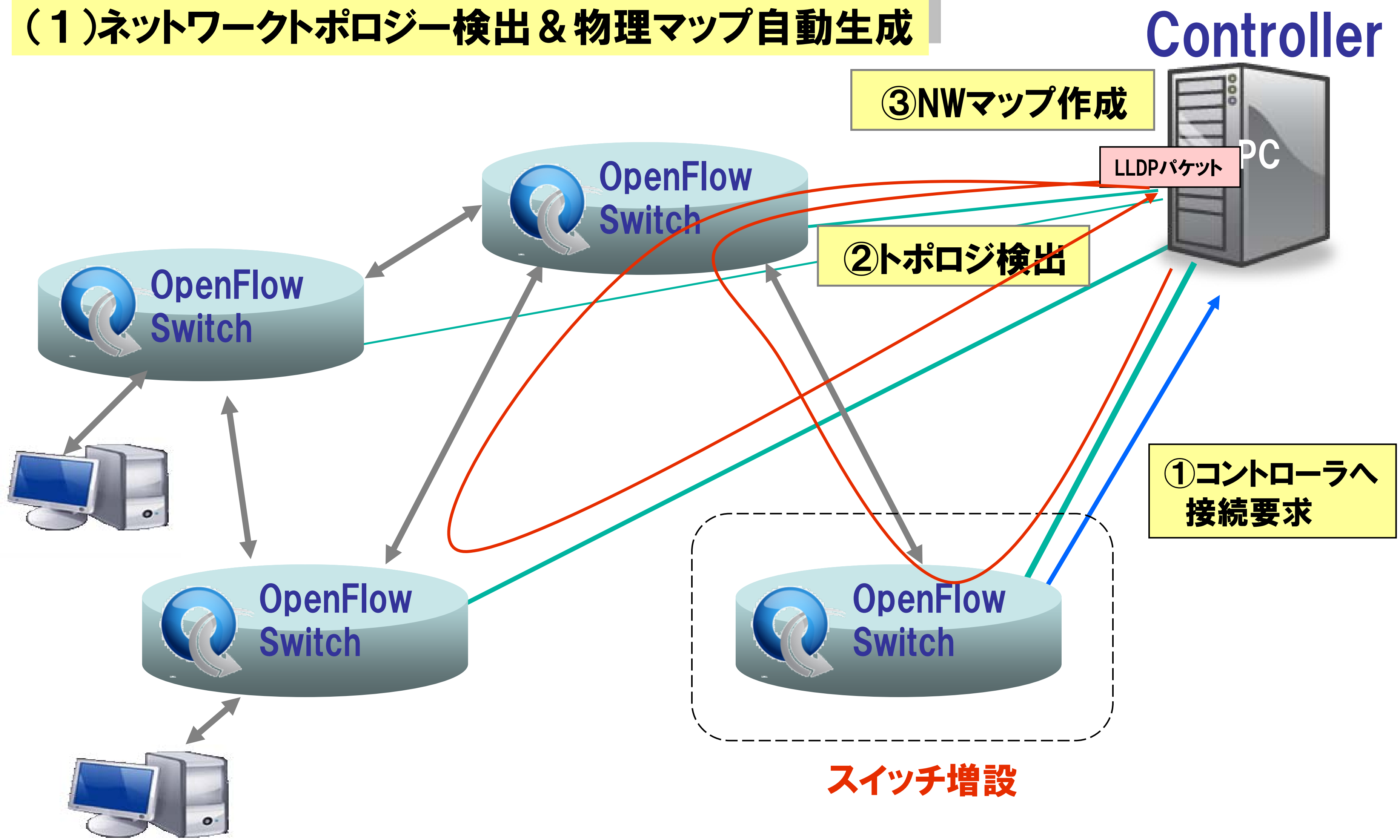
通信の終了

➢通信が終わりフローのAging時間が満了するとスイッチからフローが削除され、そのフローの最終統計情報と共にコントローラにフローがAgeoutしたことが通知される。



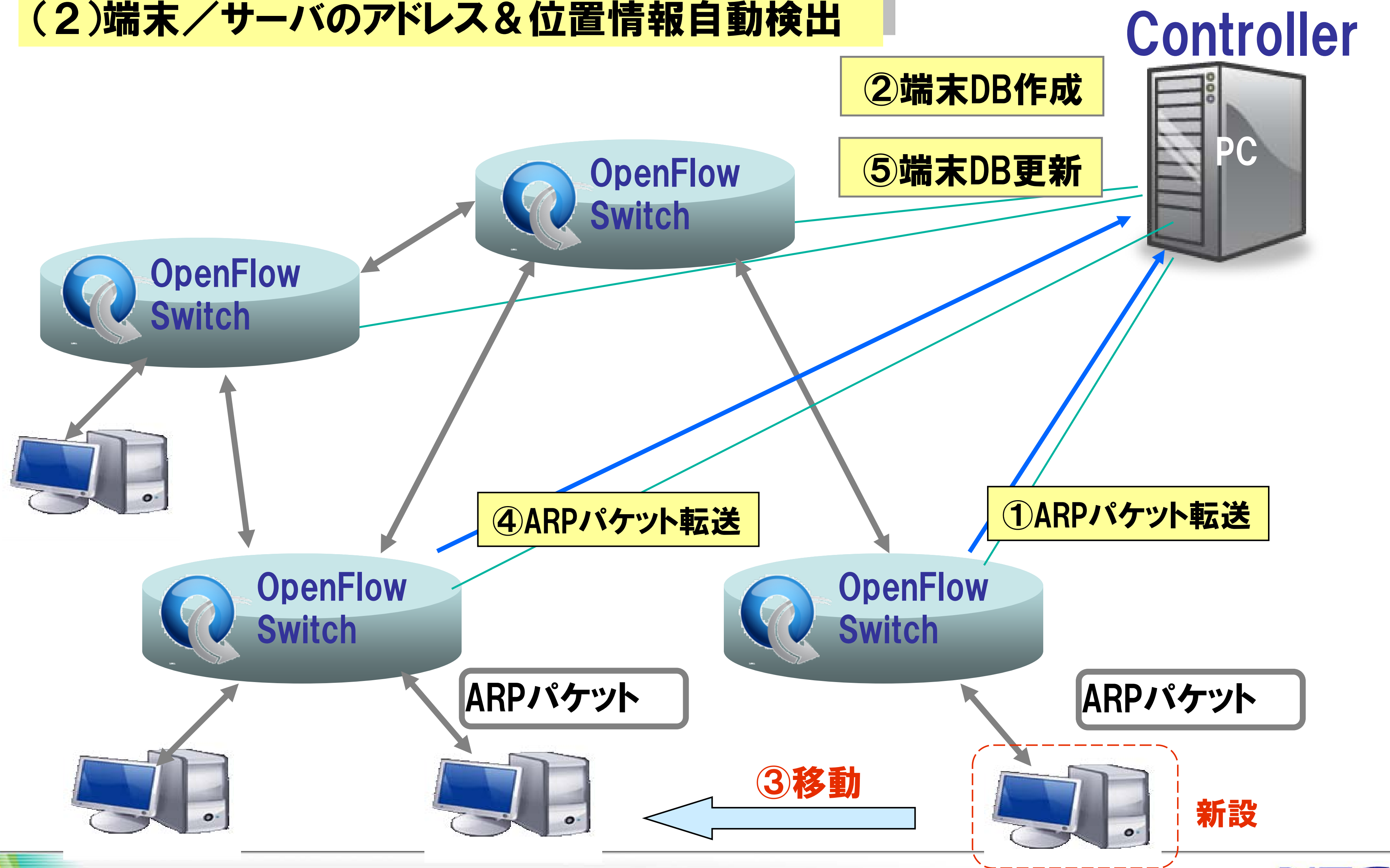
OpenFlowのシステム動作(1/4)

(1)ネットワークトポロジー検出＆物理マップ自動生成



OpenFlowのシステム動作(2/4)

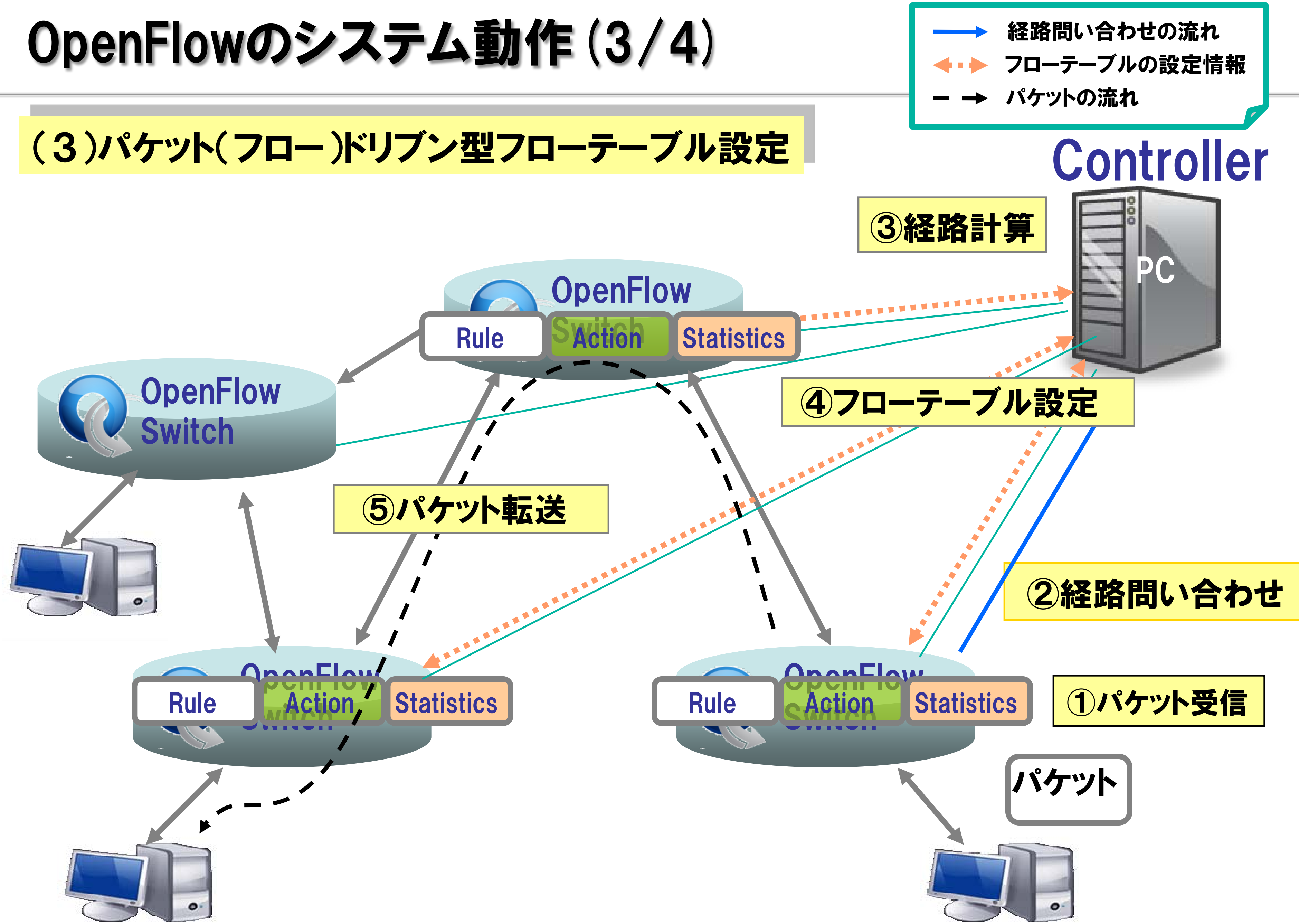
(２)端末／サーバのアドレス＆位置情報自動検出



Empowered by Innovation NEC

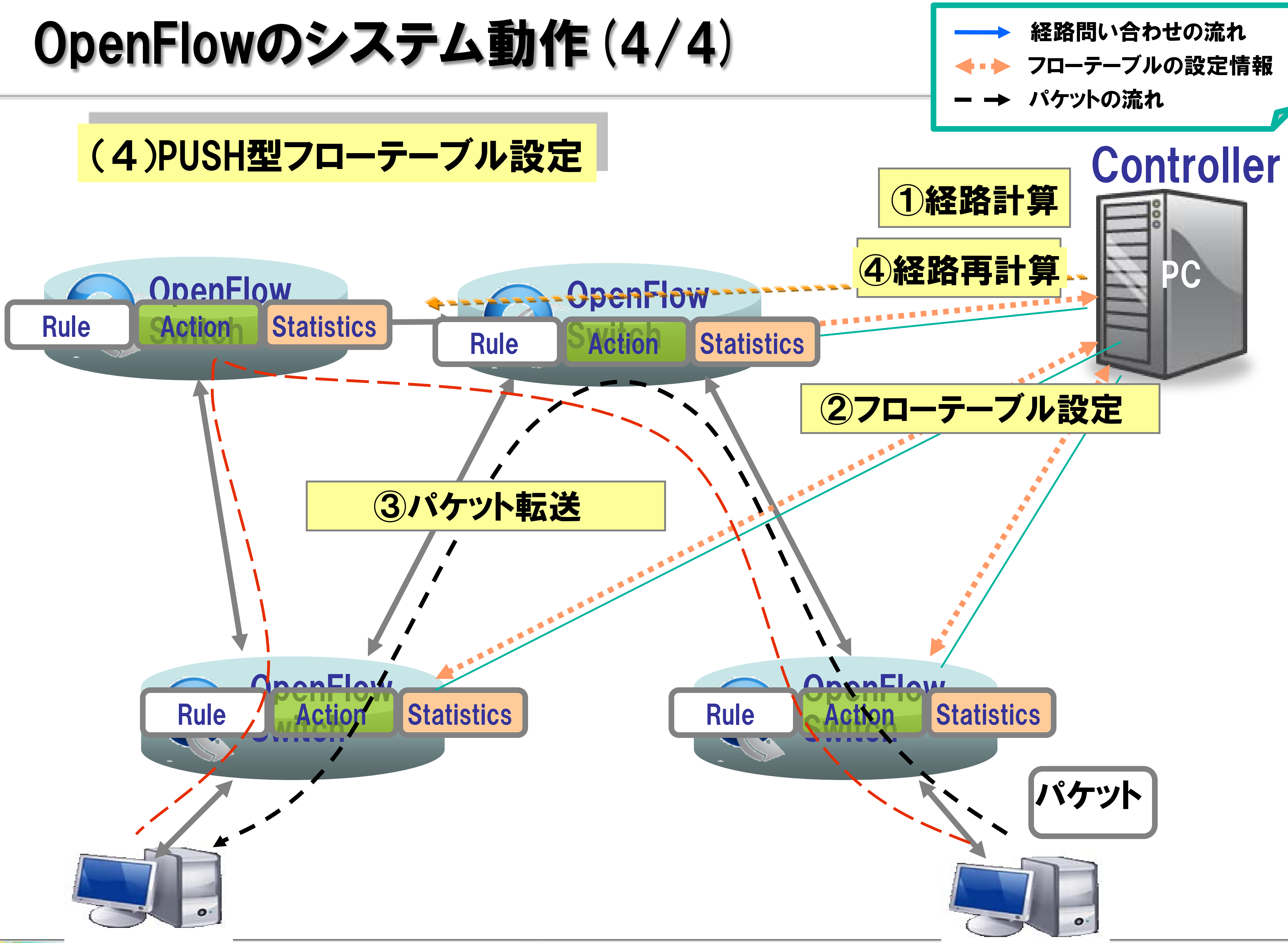
OpenFlowのシステム動作 (3/4)

（３）パケット（フロー）ドリブン型フローテーブル設定



Empowered by Innovation NEC

OpenFlowのシステム動作(4/4)



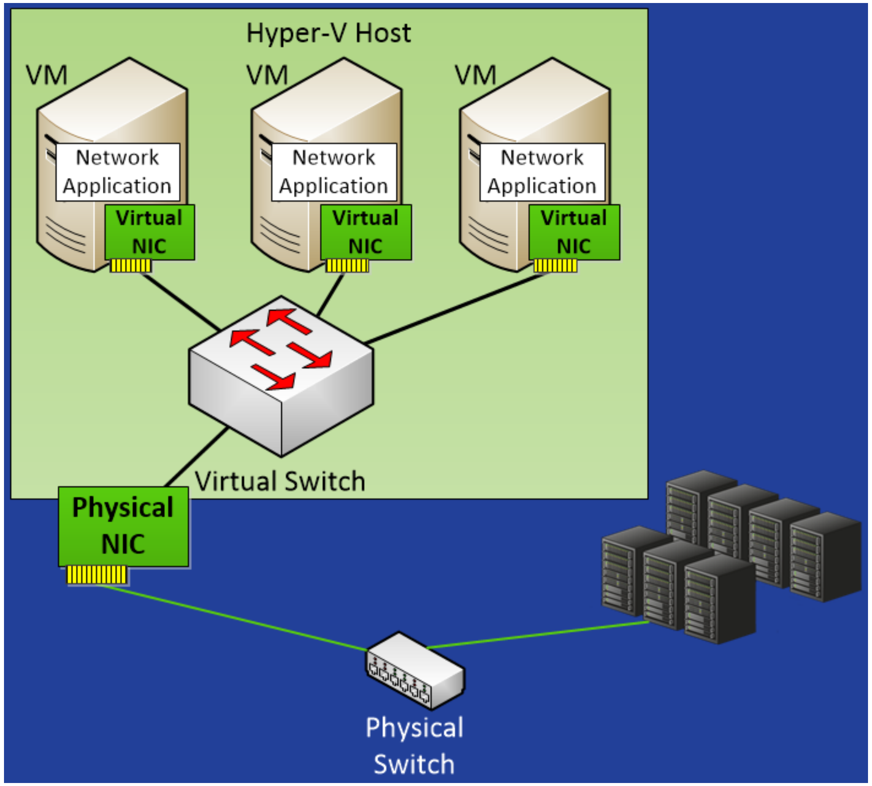
Empowered by Innovation NEC

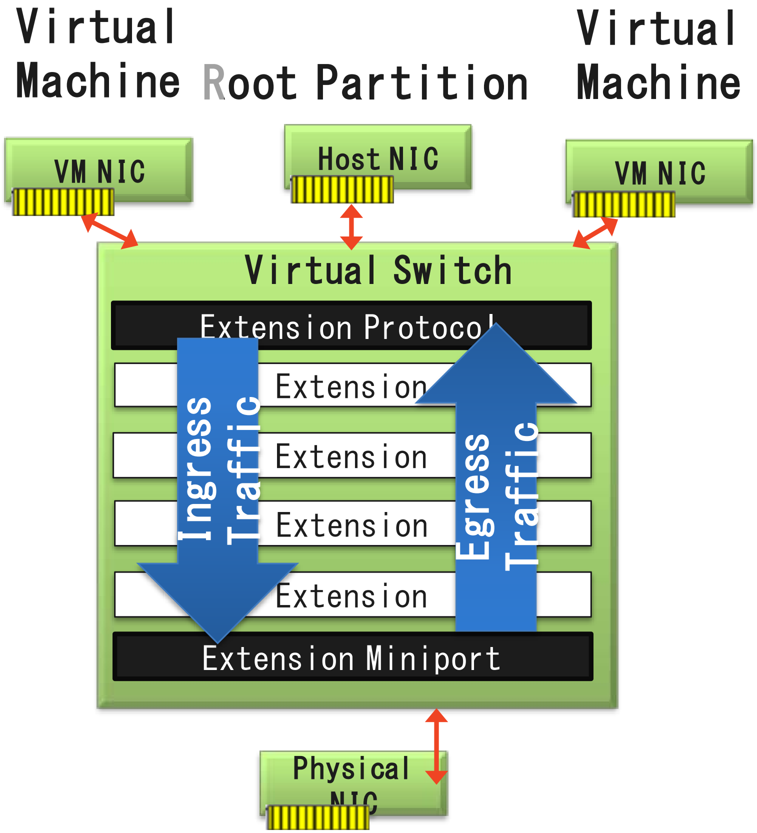
Empowered by Innovation NEC

Windows 8 HyperV OpenFlow vSwitch

Windows 8 Hyper-V Extensible Switch

- Windows 8 Hyper-V vSwitch環境上のExtension StackにOpenFlow vSwitchを実装
  - Windows Build2011で発表 & NEC HyperV vSwitch & Controller Demo
  - http://channel9.msdn.com/Events/BUILD/BUILD2011/SAC-559T

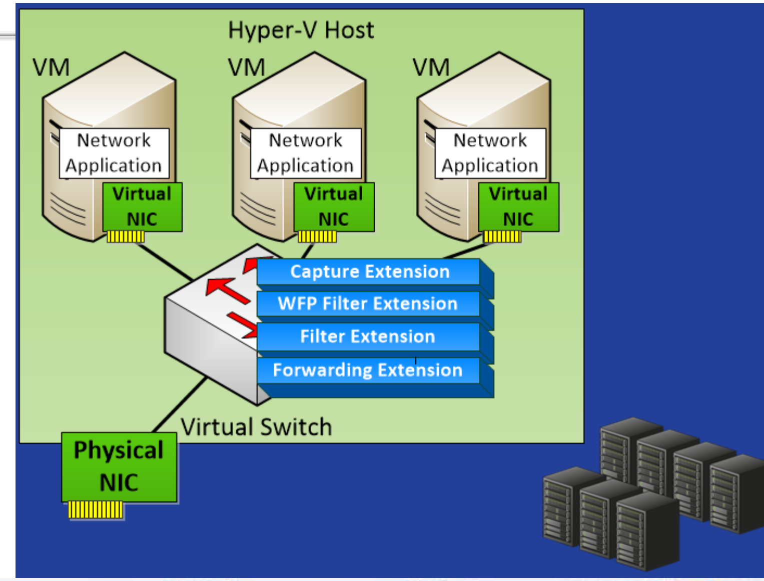


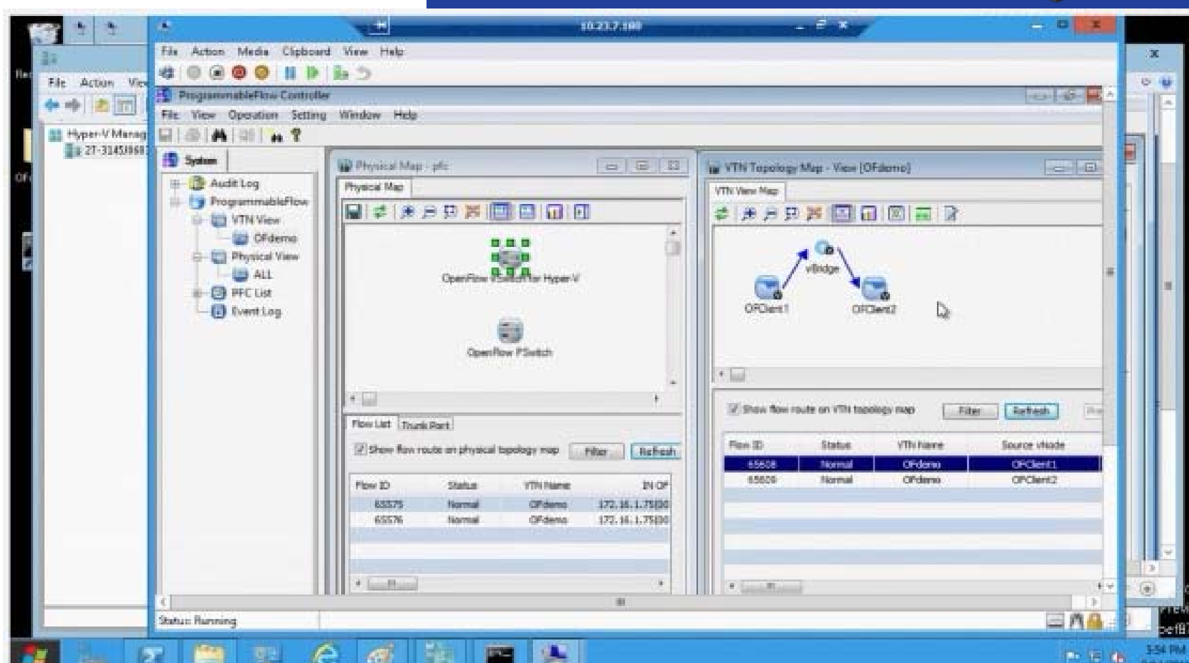


Empowered by Innovation NEC

OpenFlow vSwitch & ProgrammableFlow Controller Demo

- HyperV Extensible Switch上へ、OpenFlow vSwitch実装 & Demo
- NEC ProgrammableFlow Controller上でOpenFlow vSwitch & 物理Switch (PF5240)とを統合管理。



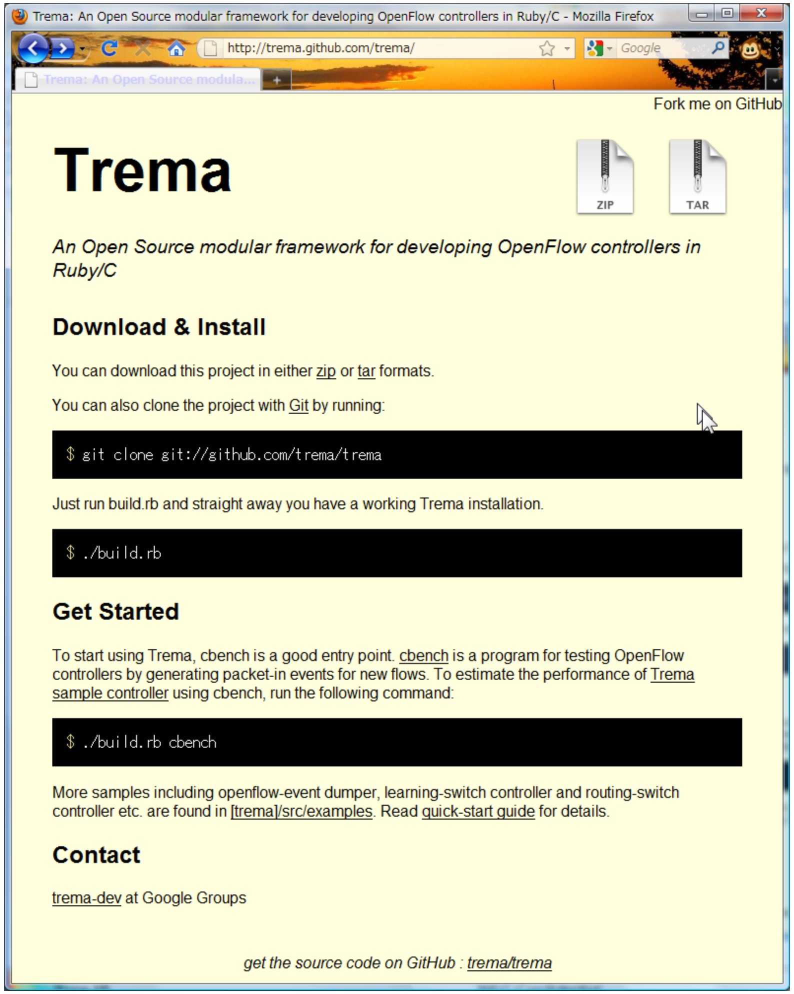


Empowered by Innovation NEC

OpenFlow controller platform “Trema” - Cloud network communityへ向けたOpen Source活動

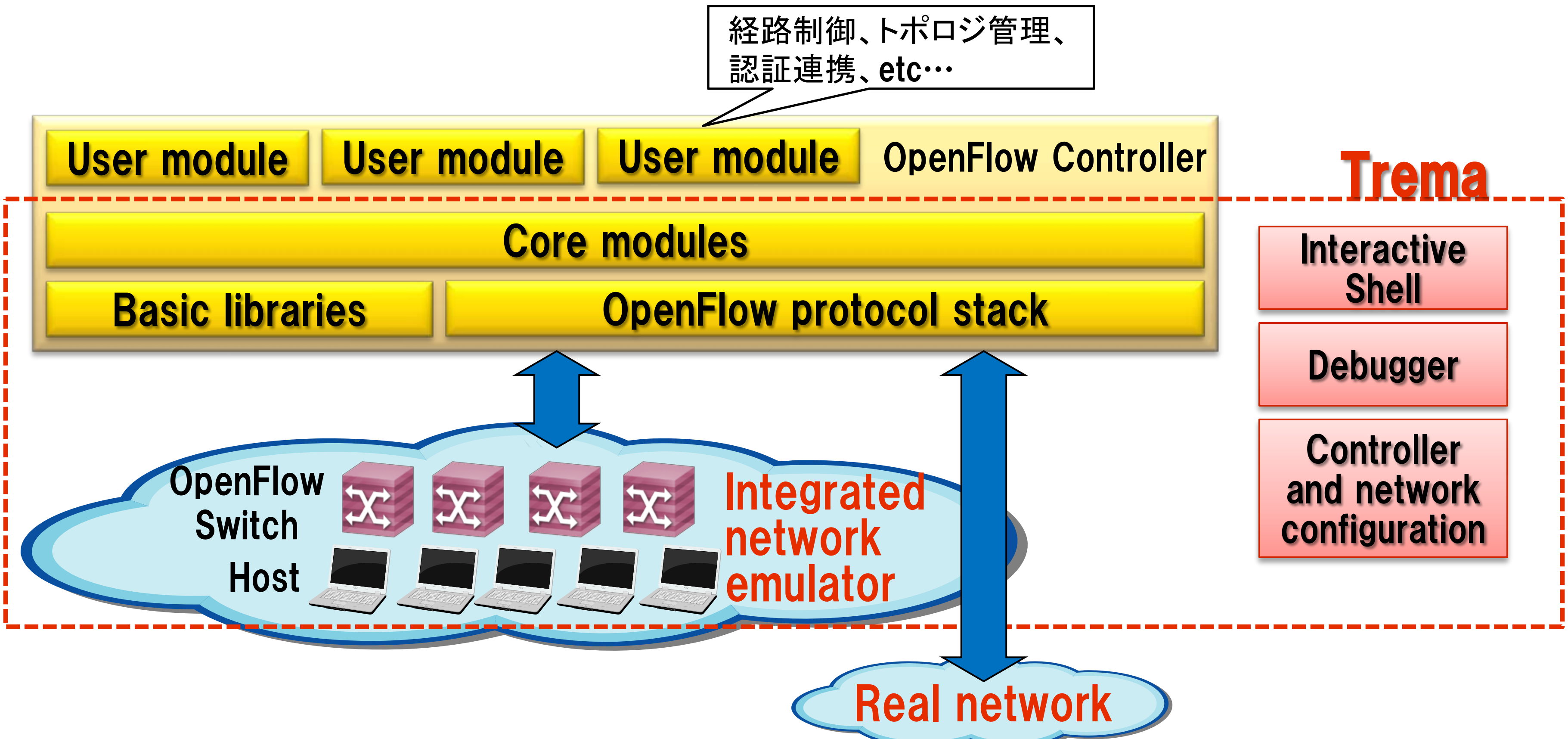
Open Source OpenFlow Controller: Trema

- フリーソフトウェア（GPLv2）“Trema”として、2011年4月に公開
  - レポジトリ https://github.com/trema/
  - ML trema-dev@googlegroups.com
  - Twitter @trema_news
- OpenFlowコントローラを作るためのプラットフォーム
  - 研究者や開発者向けプラットフォーム
  - ≠商用コントローラ
- OpenFlow開発にかかわる統合環境
  - コントローラとネットワークエミュレータの統合によるシームレスな開発環境
  - TDD（Test Driven Development）フレームワーク
  - CとRubyでの開発が選択可能



Trema概要

- Trema本体 + ユーザ開発アプリケーション ➔ OpenFlowコントローラ
  - Trema本体 @ https://github.com/trema/trema
  - 公開アプリケーション @ https://github.com/trema/apps



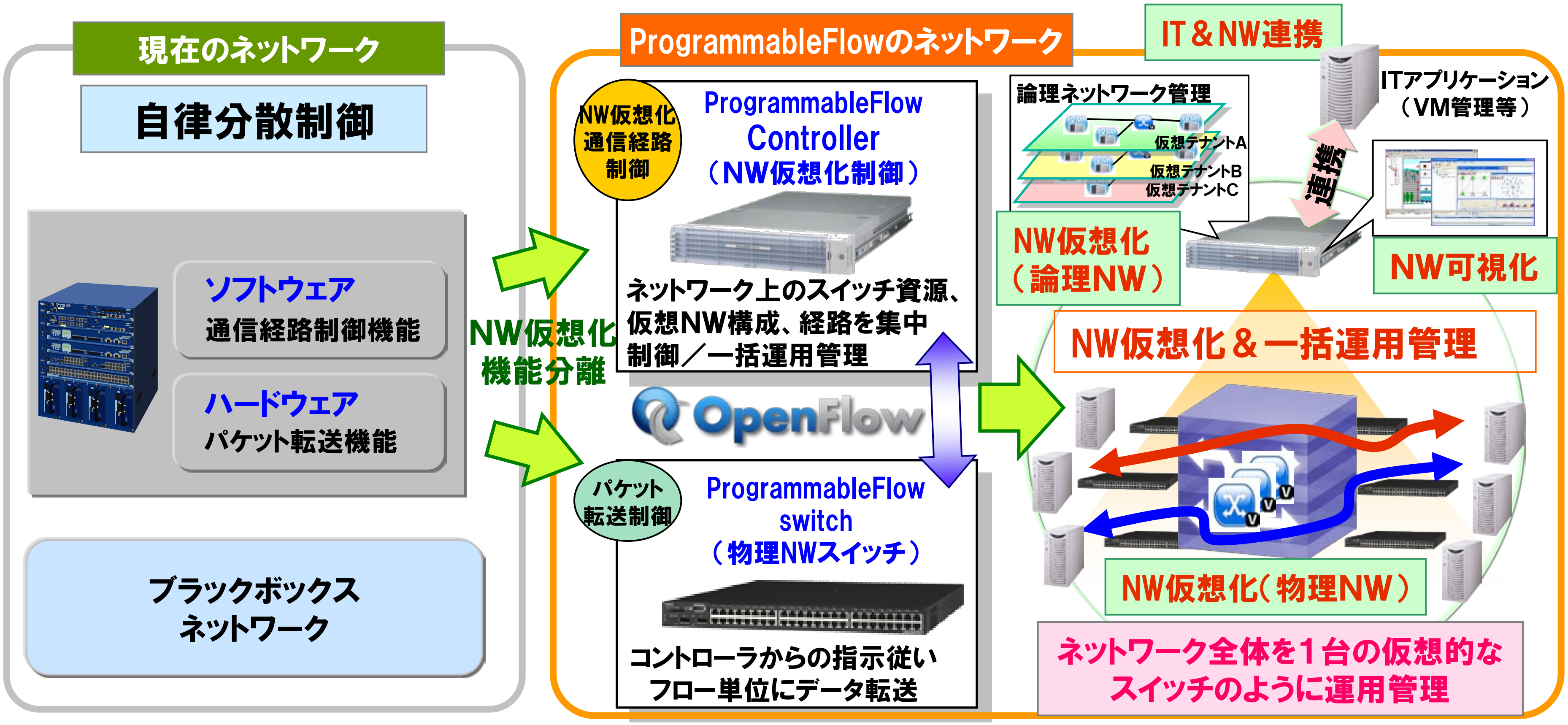
データセンター／クラウドにおける ProgrammableFlowソリューション

ProgrammableFlowによる次世代データセンターソリューション

■ ProgrammableFlowは、次世代ネットワーク技術である OpenFlow をベースに、新たなクラウドデータセンターネットワークソリューションを提供いたします。

ProgrammableFlow=OpenFlow+ NWシンプル化+ NW可視化+ NW仮想化+ IT-NW統合連携

→ クラウドデータセンターのCAPEX、OPEXを大幅に削減。



Empowered by Innovation NEC

ProgrammableFlow新製品リリース

2011年4月販売開始

http://www.nec.co.jp/datanet/p_flow/index.html

ProgrammableFlow・Controller (PFC)

UNIVERGE PF6800

UNIVERGE PF6800

- OpenFlow 1.0準拠
- OpenFlow技術により構築されたネットワーク制御情報の管理を実現するハードウェアアプライアンス製品
- フロー(MACアドレス、IPアドレス、ポート番号などの送信元及び宛先アドレスの組み合わせ)による、きめ細かな経路制御、トラヒック制御、監視を実現する
- 冗長構成対応、商用利用時は冗長必須。

ProgrammableFlow・Switch (PFS)

UNIVERGE PF5240

UNIVERGE PF5240

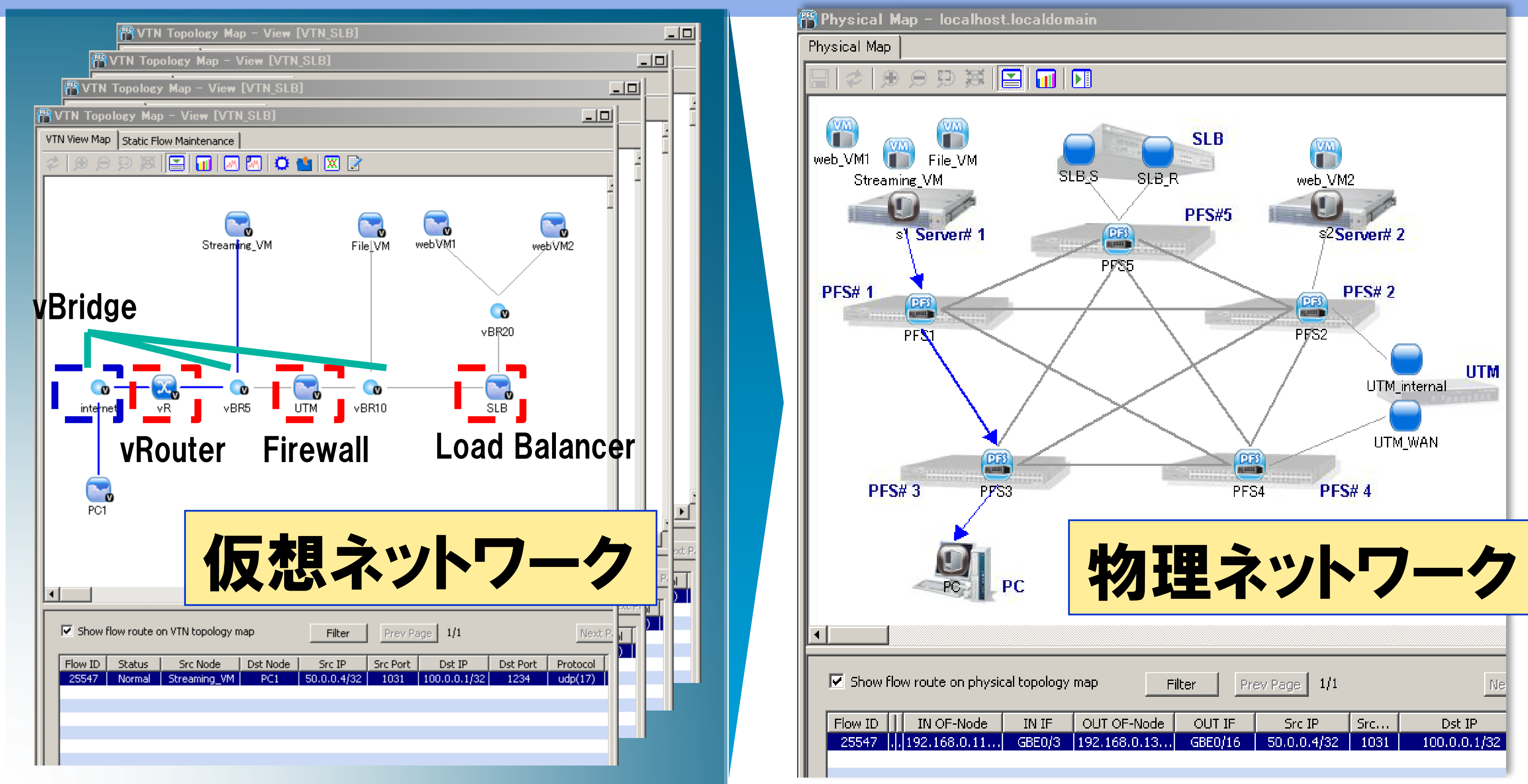
- OpenFlow 1.0準拠
- ハードウェアによる「OpenFlow」機能のフローエントリ検索、転送アクション実行によるフルワイヤレートでのパケットフォワーディングを実現
- 1GbE x 48 + (10GbE (SFP+) or 1GbE (SFP)) x 4収容
- 多フローEntry対応 (12 Tuple, wire-rate)
- 電源機構は非冗長、冗長の選択可能

BEST OF INTEROP 2011
Infrastructure
PRESENTED BY: InformationWeek analytics

ネットワーク可視化（物理網、仮想網構成）

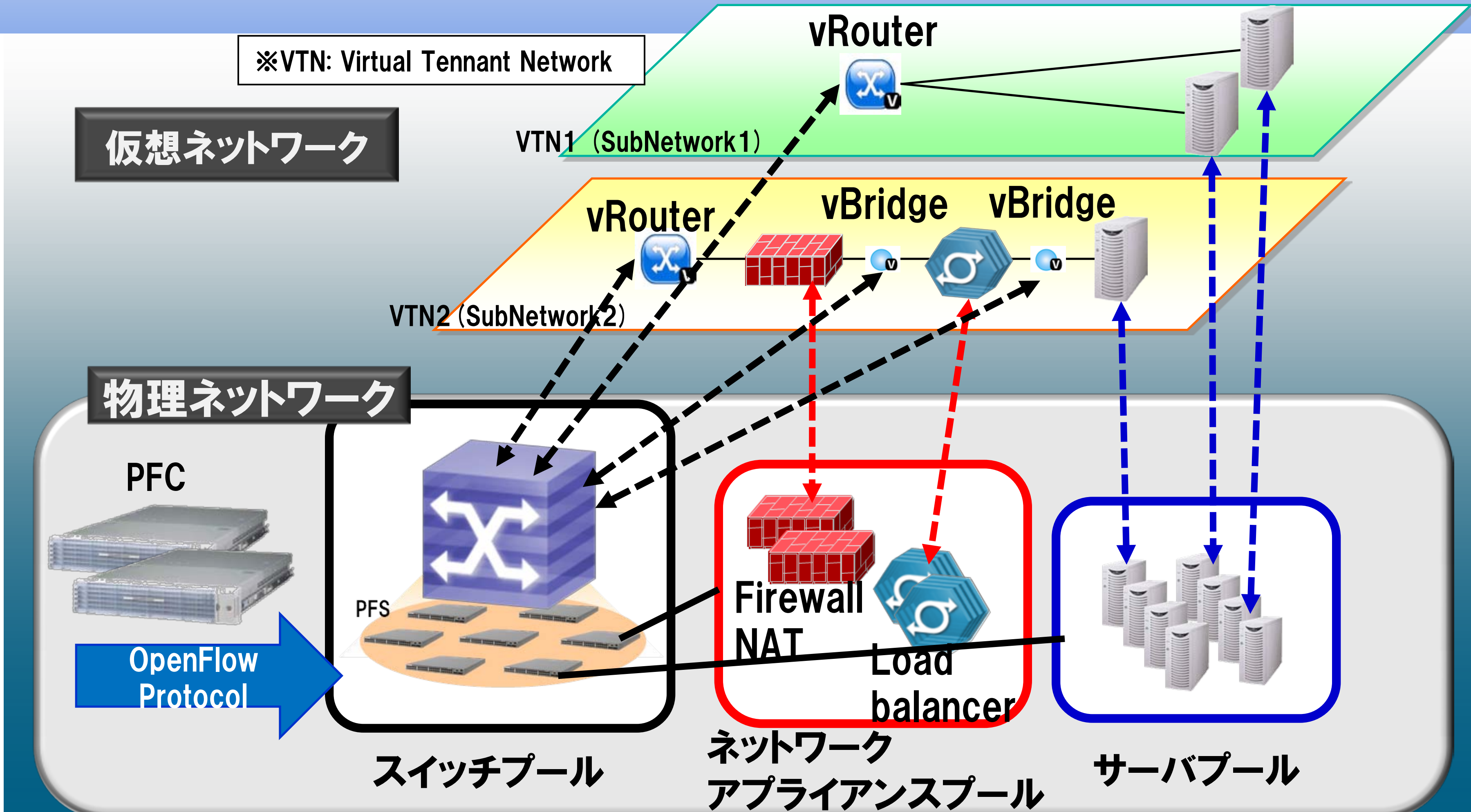
ネットワークを物理NW、仮想NWで管理、可視化が可能

- L2/L3機能からL4/L7機能まで仮想網上で設計、経路制御（CLI ）
- 物理NW、仮想NW上でトポロジー、通信フローの可視化（GUI）
- ネットワーク設定/運用維持コストを削減（仮想NW設定は物理NWに非依存）



ネットワーク仮想化 #1 （L2/L3レイヤからL4/L7まで）

vRouter, vBridge, ネットワークアプライアンス, サーバ/端末を仮想NW上で統合設計 （ L2/L3機能からL4/L7機能までのネットワーク論理設計 ）



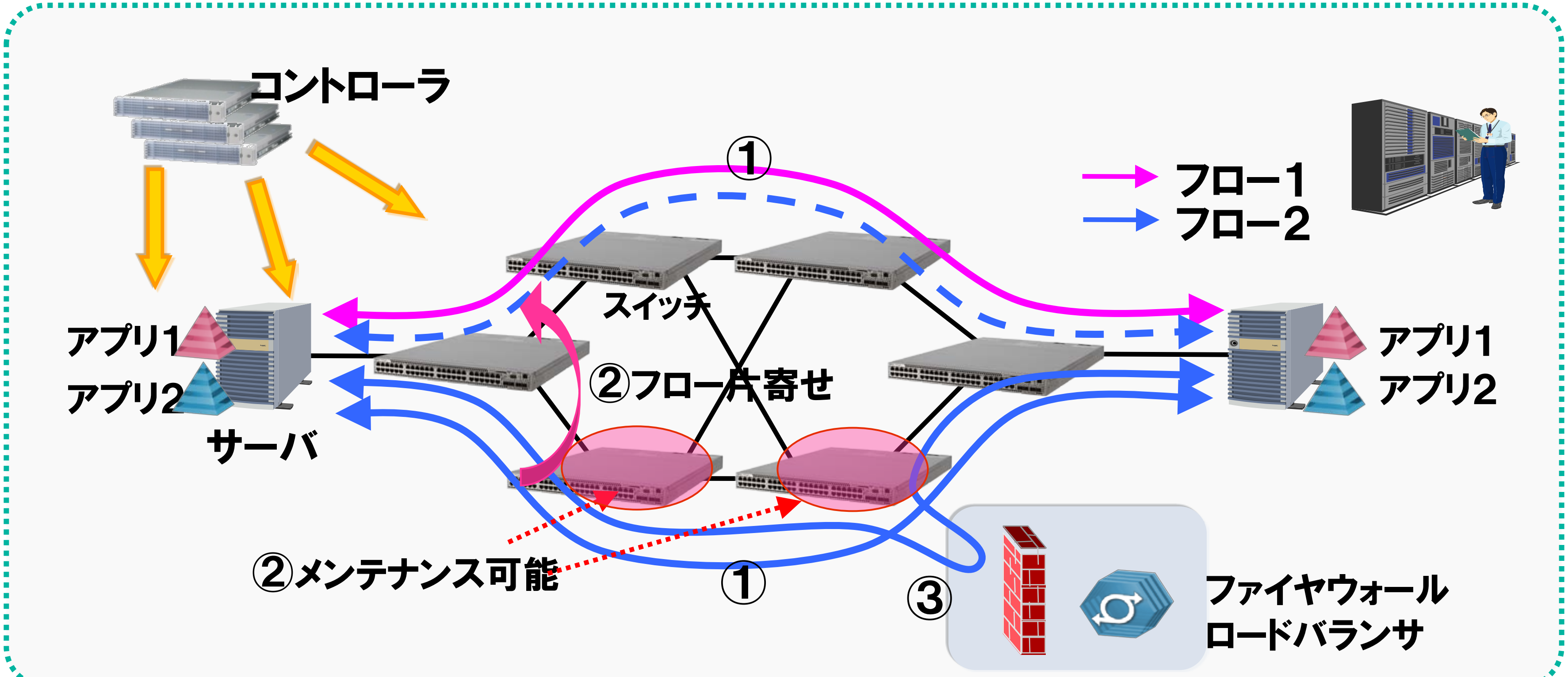
ネットワーク仮想化 #2: トラフィック経路制御の高度化

■ オープンフローによるトラヒック経路制御の高度化

①フロー単位の経路制御によるネットワーク帯域の有効利用(マルチパス)

②フローの片寄せによるネットワーク機器のメンテナンス性向上

③NAT, Firewall, Load balancer等のネットワークアプライアンスプールを配備し、特定フローの通過設定可能(WayPoint機能)。高価なアプライアンスを適宜トラヒックOffloadすることでトータル設備コストを削減。



Empowered by Innovation NEC

ProgrammableFlowと既存LAN／WANとの接続

＜上位ルータとの接続＞

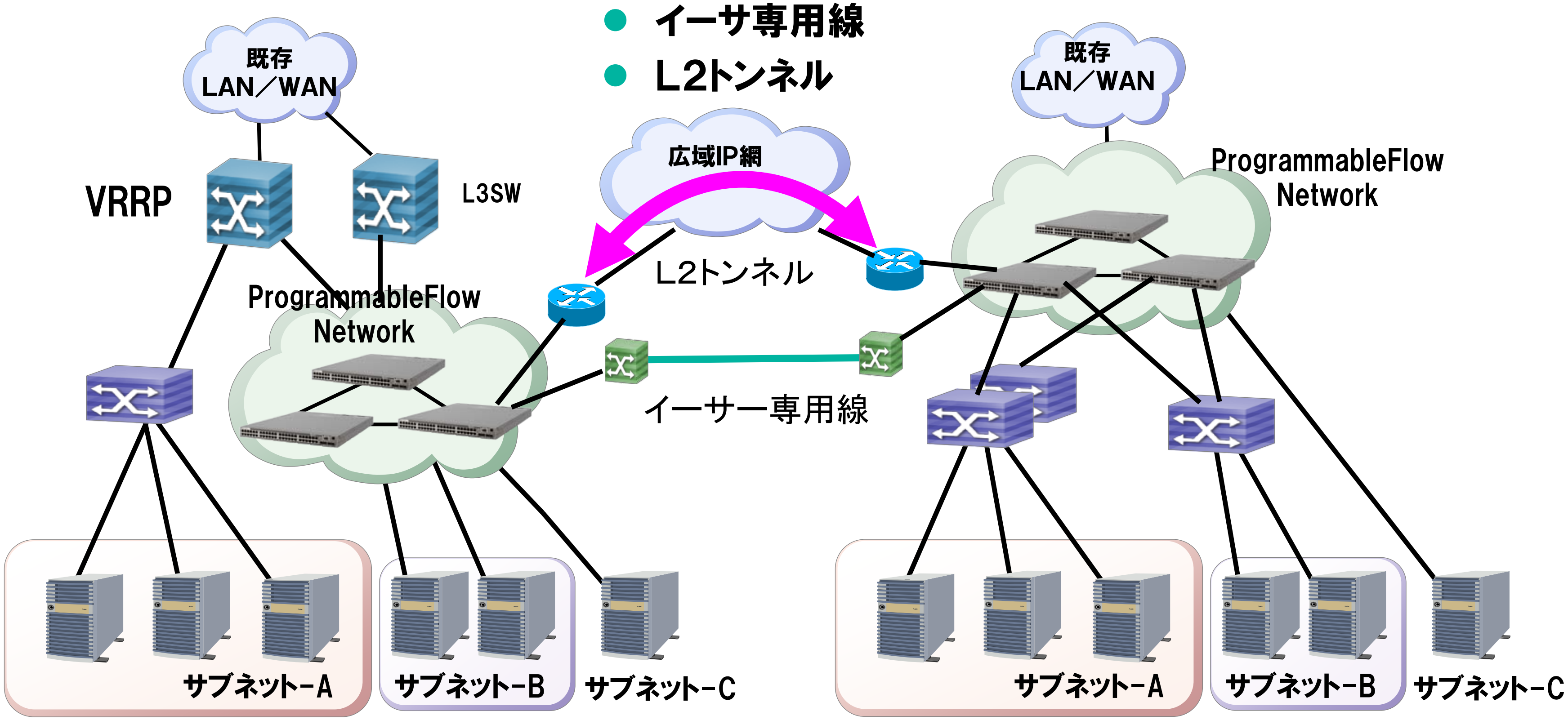
- 既存ルータやL3スイッチから、1つの枝として OpenFlowネットワークをぶら下げる。
- 上位ルータとの間はVRRP/LAGで冗長接続をし、static routeで接続する。

＜下位スイッチとの接続＞

- 既存ネットワーク配下のサーバを OpenFlowネットワーク配下に移動することで移行が可能。
- 下位スイッチの間はLAGで接続する。

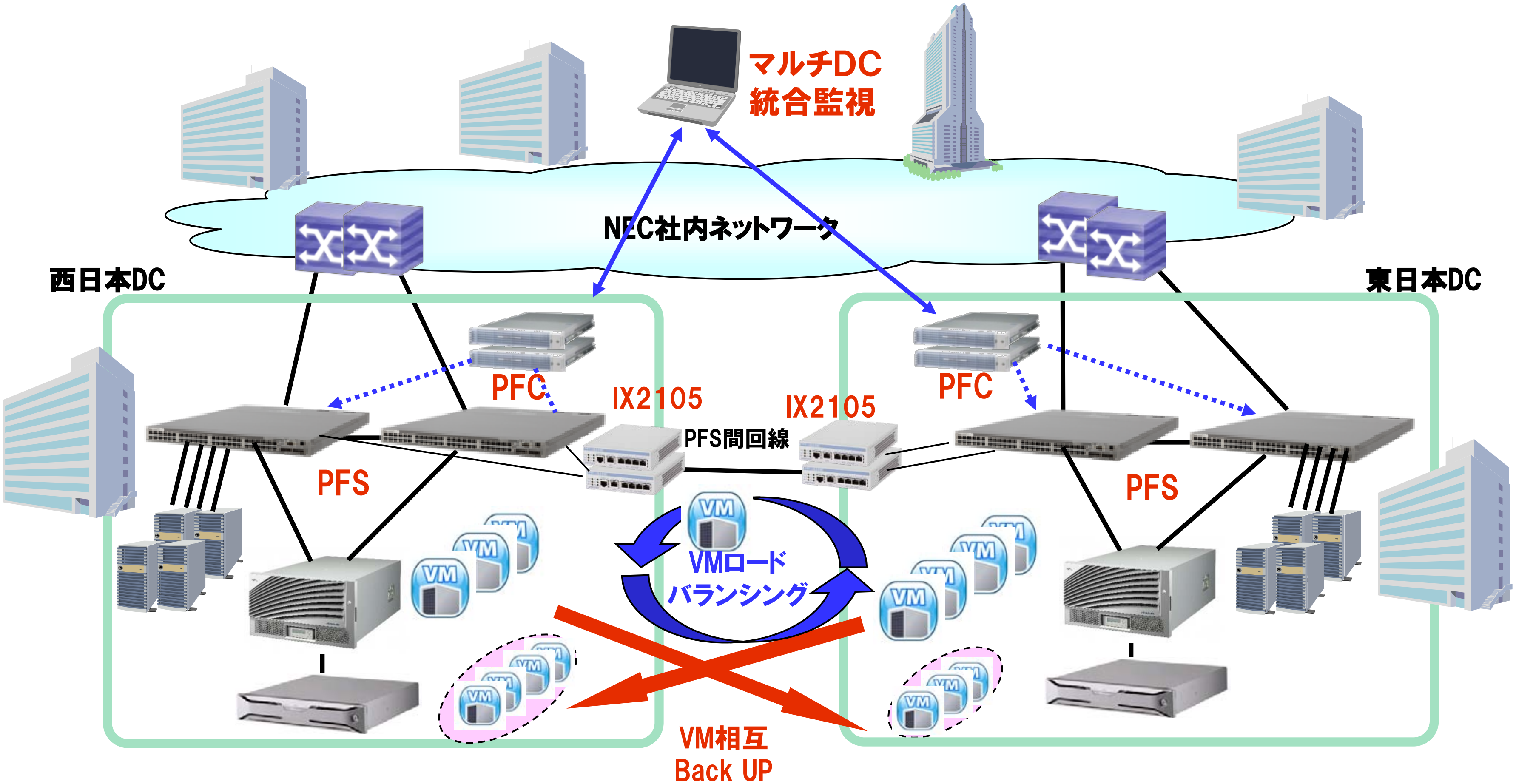
＜WAN間接続＞

- イーサ専用線
- L2トンネル



NECソフトウェアファクトリでのProgrammableFlow適用

- 国内に散在している事業所のITリソースをプログラマブルフローを導入したDCに集約。ソフト開発者が効率的に活用
- DC間で負荷に応じたロードバランシング

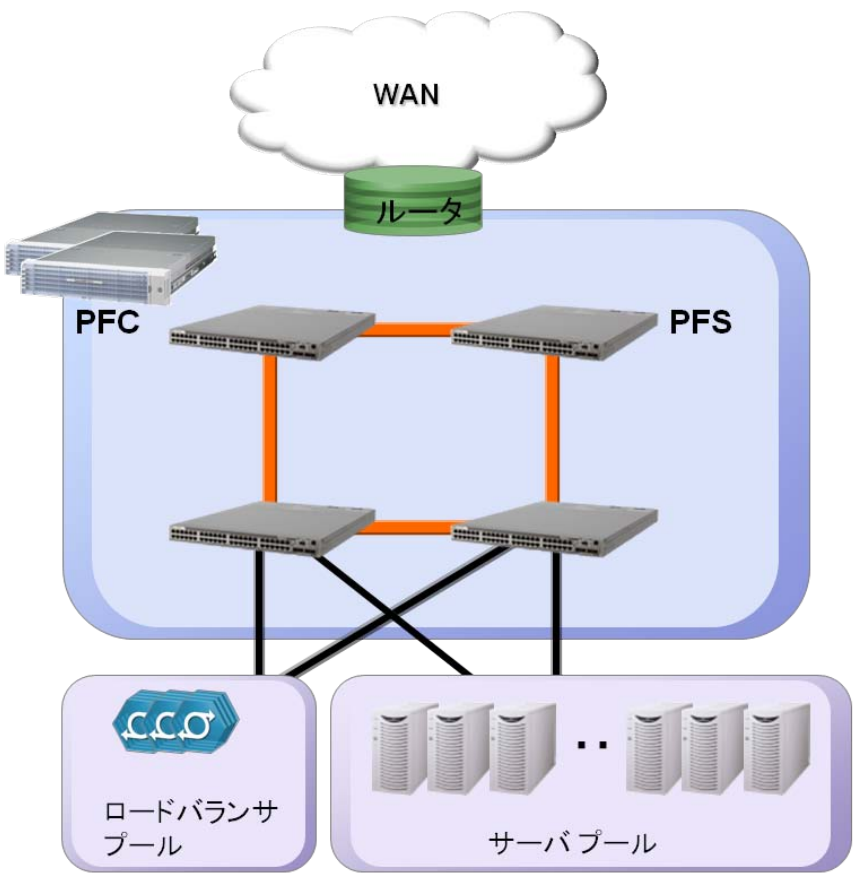


導入事例

日本通運株式会社様

世界初、業務システムへの導入

日本通運様　構成イメージ



※ネットワーク可視化の機能については次期バージョンにて強化予定

プログラマブルフロー導入による効果

- ネットワーク設定の変更には1回あたり100～200万程度のコスト(2010年度は年間3回実施)がかかっていたが、プログラマブルフローの導入によって、**物理構成の変更が容易となったため**、自社の社員で作業が可能になり、設定・変更費は実質無料。
- ネットワーク構築のリードタイムが、通常2ヶ月のところ10日で可能となった。

プログラマブルフローへの今後の期待

- 全システムの標準化の基盤としての利用
  パブリッククラウドとプライベートクラウドを共有し、利用者が意識しないで利用できる環境を、プログラマブルフローで構築したい

導入事例

Genesis Hosting 様（シカゴにあるサービスプロバイダ）

導入の背景

- ネットワークの複雑化が極端に進み、大規模ネットワーク向けの新たな戦略が必要
- 「低価格かつ高可用」のサービスが他社との差異化方針
- 利用可能なIPアドレス数が減少

プログラマブルフロー導入による効果

- シンプルで扱い易いネットワークとなり大規模な拡張も可能
- Genesis Hostingが目指す99．999％の可用性実現を支援
- ネットワーク管理時間が100時間／週も短縮
- IPアドレス利用数が最大60％削減
- マルチテナント仮想ネットワークサービス提供により収益UP

ProgrammableFlow製品情報と問い合わせ先

ProgrammableFlow製品概要情報、カタログ情報:

http://www.nec.co.jp/datanet/pflow/

企業・官公庁・通信事業者のお客さま　導入前のご相談・お問い合わせ窓口:

https://www.nec.co.jp/cgi-bin/contact/input.cgi