

IT/ネットワーク統合制御プレーン 向けネットワークOS

下西 英之

要 旨

スイッチの機能を外部の制御サーバから制御できるOpenFlow技術が注目され始めています。弊社は、この制御サーバ上に仮想化された制御プレーンを構築することで、仮想インフラストラクチャを構築する方法を提案します。本稿では、仮想インフラストラクチャ全体のアーキテクチャ、及び制御サーバの統合プラットフォームであるネットワークOSについて述べます。

キーワード

●OpenFlow ●インフラ仮想化 ●ネットワーク仮想化 ●新世代ネットワーク ●FutureInternet

1. はじめに

インターネットが登場し、商用解放されて以来、インターネットはさまざまな用途で用いられるようになってきました。当初想定されていたようなコンピュータ間の通信としてWebサービスなどに用いられるだけでなく、音声通信や映像配信などにも用いられ、さまざまな技術がインターネット技術上に追加されてきました。そのため、シンプルであったレイヤ構造が複雑化していき、システムの構築・維持・管理のコストが増加しただけでなく、ネットワークへの新しい機能の追加やサービスの追加が困難となりました¹⁾。新しい機能を開発したとしても、インターネット上で動作しているルータに機能追加し、新規サービスを開始するためには非常に長い時間とコストが必要となるためです。

このようなイノベータ不在となりつつあるインターネットインフラを再考するため、米国のNSF FINDや欧州FP7での議論、日本での新世代ネットワークの議論など、FutureInternetに関する議論が近年非常に活発化してきています¹⁻³⁾。また、そこで生まれる新しい技術を実証するための環境として、GENI⁴⁾などの大規模テストベッドが構築・運用され始めています。

以上のような背景のもと、我々は新しいインターネットインフラのアーキテクチャの検討を行っています^{5, 6)}。我々のゴールは、IP技術のようにone fits allなインフラを実現することではなく、プログラマビリティと仮想化技術によって新しい仮想インフラを容易に構築でき、それによって多様性のあ

るインフラは、特定の技術に長年依存することなく、その時々で形を変え進化していくことが可能であると期待します。

提案するアーキテクチャは、OpenFlow技術⁷⁾を用いてネットワーク機能のプログラマビリティを、スイッチ側ではなく、制御サーバ側に持つことを基本とします。また、制御サーバではOpenFlowだけでなく無線や仮想マシン（VM）などのさまざまなインフラ資源の制御も統合的にを行い、仮想インフラの構築を制御サーバ上で行えるようにします。本稿では、仮想インフラを容易に構築・運用するための制御サーバ上のソフトウェアプラットフォームとして“ネットワークOS”を提案します。提案するネットワークOSは、NOX⁸⁾に代表されるようなOpenFlowコントローラとしての役割だけでなく、仮想インフラ構築のため、また、統合制御プレーンとしてOpenFlowだけでなく仮想マシンやその他の資源を制御するため、物理資源を仮想資源に切り分けてサービスとして制御モジュール群に提供する機能や、仮想インフラごとに制御モジュール群を隔離して管理する機能を持つことが特徴です。

以下本稿では、関連する研究動向のまとめ、提案する仮想インフラストラクチャのアーキテクチャ、そしてそれを実現するネットワークOSについて述べます。最後にネットワークOSの試作システムとその上のアプリケーション実装に関して簡単に述べます。

2. 関連動向と要求条件

新しいネットワークインフラのアーキテクチャを検討するに当たり、関連する研究領域ではさまざまな提案がなされて

います¹⁻⁹⁾。本稿では個々の研究については述べませんが、これらに共通する要求条件として以下の4点を挙げます⁹⁾。

(1) プログラマビリティ

ユーザ独自の機能を持った新しいインフラを構築するためには、ネットワークを含めたインフラ全体の機能をソフトウェアのようにデザインできることが必要です。OpenFlowはスイッチのプログラマビリティを制御サーバ上で実現するための技術であり、制御サーバ上での高いプログラマビリティを提供します。

(2) 仮想化

実インフラにおけるプログラマビリティを保証するためには、コンピュータにおける仮想マシンと同様に、複数のユーザが物理インフラを共有し、なおかつ仮想的には独自のインフラとして自由な構築・運用ができることが重要です。

(3) オープン化

プログラマビリティや仮想化は誰でも容易に利用できるようオープンでなければなりません。コンピュータではハードウェアを容易に使いこなすためにOSが進化し、その上で自由にイノベーションが可能であったのと同様に、ネットワーク機器においても同様の仕組みが重要です。

(4) モジュール化

いろいろなネットワーク機能を部品化し、ユーザが自由にネットワークインフラをカスタマイズできることが重要です。さまざまなユーザが開発したモジュールを自由に組み合わせることで新しい機能を容易に実現できること（マッシュアップ）は、イノベーションの活発化に重要です。

それぞれの制御プレーン間は疎結合になっています。そのため、さまざまなシステムを要素として含むインフラ全体に対して、統一したプログラマビリティと仮想化を提供できず、仮想インフラの構築が困難となっています。

そこで、図1右のように制御プレーンを全体で統合し、この統合制御プレーンからさまざまな種類の物理インフラを統合して制御するアーキテクチャを提案します。第1章で述べた通り、プログラマビリティは制御プレーン側で持ち、この上で統一された仮想インフラの構築環境を提供します。

(2) インフラストラクチャ仮想化

図2に統合制御プレーンによるインフラ仮想化について示します。単一物理インフラ上に複数の仮想インフラを実現するため、物理インフラからスライスとして切り出された仮想資源群（例えば仮想スイッチや仮想マシン）を仮想インフラごとに用意し、それぞれ独自の制御プレーンを組み合わせることで、それぞれアプリケーションに対して最適化された仮想インフラを構築することができます。例えば、

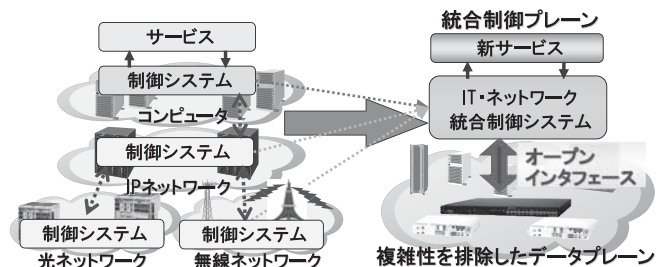


図1 統合制御プレーン

3. 提案する仮想インフラのアーキテクチャ

本章では、第2章で述べた要求条件をもとに、提案するアーキテクチャについて説明します。

(1) 統合制御プレーン

図1は統合制御プレーンの考え方を示す図です。一般的にネットワークシステムは、実際にデータを処理するデータプレーン（図中装置）と、それを制御する制御プレーン（図中制御システム）に分かれています（それを管理する管理プレーンも存在します）。それぞれのシステムは、図1左のように、光トランスポートや無線ネットワークなどのシステムごとに独立したものとなっており、基本的にそれ

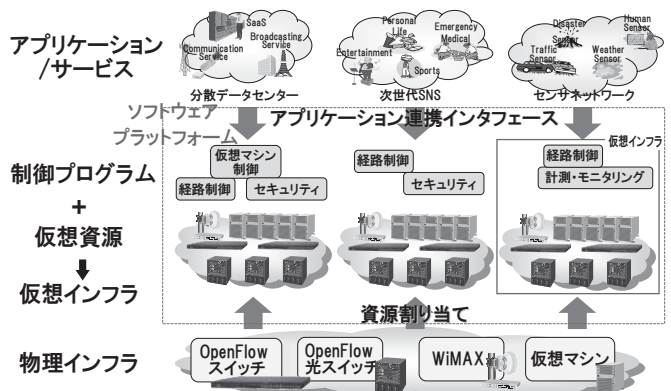


図2 仮想インフラの構築

センサネットワーク用の仮想インフラが必要であれば、そのために必要な仮想マシンや仮想スイッチなどの仮想資源を物理インフラから切り出し、センサネットワーク用の経路制御や仮想マシン制御などの制御プログラムを統合制御プレーンに組み込むことで一つの仮想インフラを形成します。

4. ネットワークOS

以上のような仮想インフラを構築するための統合制御プレーンのソフトウェアプラットフォームがネットワークOSです。図3に示すように、OpenFlowスイッチなどの物理資源を制御する制御サーバ上のミドルウェアであり、経路制御やネットワーク計測などのさまざまな制御モジュールに対して仮想化された物理資源を制御するためのオープンなAPIを提供します。また、OpenFlowコントローラとしての役割だけでなく、仮想インフラ構築のため、また、統合制御プレーンとしてOpenFlowだけでなく仮想マシンやその他の資源を制御するため、図4のように制御モジュールとサービスが階層的になった制御構造を持ちます。このような構造によって、複数のモジュールを組み合わせることで容易に仮想インフラを構築することができます。

(1) Network OSが提供するサービス

ネットワークOSが提供するサービス（以下単にサービス）とは仮想スイッチや仮想マシンなどの仮想化された物理資源だけでなく、ネットワークトポロジやフローパスなどの制御モジュールによって生成された論理資源も含み、それ

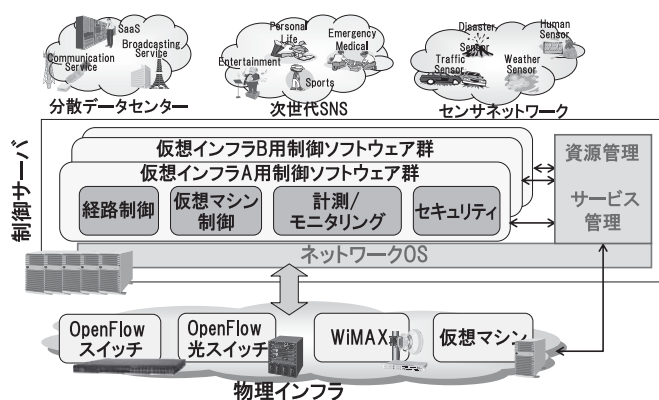


図3 ネットワークOS

らの資源に対する操作を与えるものと定義します。例えば、“OpenFlowスイッチ”というサービスは、仮想スイッチという仮想資源と、これに対してフローを設定するためのAPIを与えます。また、“フローパス”というサービスは、フローに対するパスの集合という論理資源と、これに対してパスの生成削除などの操作のためのAPIを与えます。ネットワークOS上ではこれらのサービスに対して一意な名前空間を与える機能を持ちます。したがって、提案するネットワークOSにおいては、サービスは以下を持つものとして定義されます。

- 1) サービスの名前
- 2) 仮想資源/論理資源に対応付けられたデータ構造
- 3) 資源を操作するためのAPI

そして提案するネットワークOSは、図4に示す通り、制御モジュールが仮想資源/論理資源をサービスとして用いて処理を行うという構造と、制御モジュールがその処理を行った結果として新たなサービスを生み出すという構造が階層的に組み合わされた制御構造を持ちます。

(2) 制御モジュール

制御モジュールは、トポロジディスカバリやパス生成などのネットワークを制御するアルゴリズムが実装された実体です。また、ネットワークだけでなく、サーバロードバランスや仮想マシンマイグレーションなどのサーバ資源を制御するものも含まれます。図4の下段に示されている通り、スイッチなどの物理資源に直接接続され、これを仮想資源と

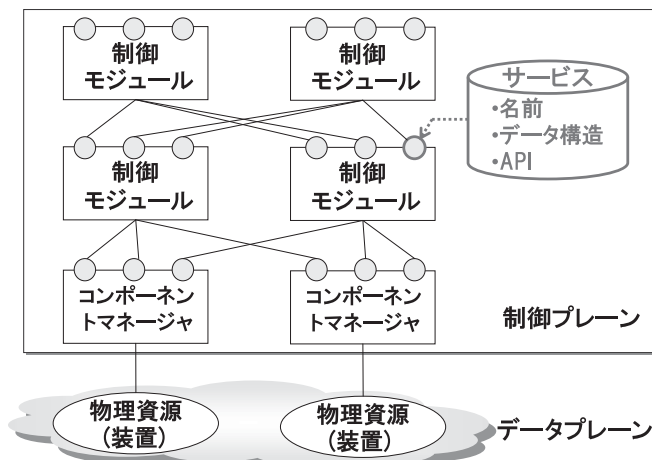


図4 制御モデルの基本構造

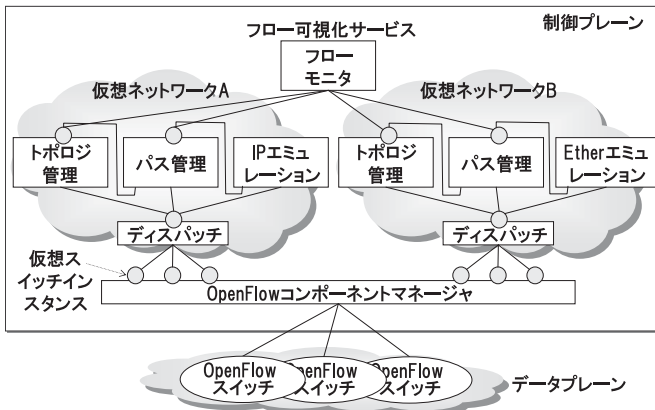


図5 仮想インフラの構築

して提供し直す制御モジュールは特別にコンポーネントマネージャと呼びます。例えば、OpenFlowコンポーネントマネージャは、TCP/SSLで複数のOpenFlowスイッチに接続され、“仮想スイッチインスタンス”という仮想資源をパス生成などのほかの制御モジュールに対して提供する役割を持ちます。

(3) 仮想インフラの構築

上記のような制御構造により、提案するネットワークOSは仮想インフラの構築を行います。例えば 図5 のように、OpenFlowコンポーネントマネージャが提供する仮想スイッチインスタンスというサービス群を基点として制御モジュールを積み上げることで一つの仮想インフラを構築します。また前述サービスを複数に切り分け、複数の制御モジュール群を構成することで、複数の仮想インフラを構築します。

5. 試作と評価

我々は提案するアーキテクチャの実証を行うため、ネットワークOS及びその上のさまざまな制御モジュール群の試作評価を行っており、本章ではその概要について説明します。

試作したシステムは 図6 に示すように、ネットワークOSを実装した制御サーバと、これらに制御されるOpenFlowスイッチ群、サーバ群で構成されます。ネットワークOS上には、Topology discovery（トポロジ管理）、Path manager（経路の設

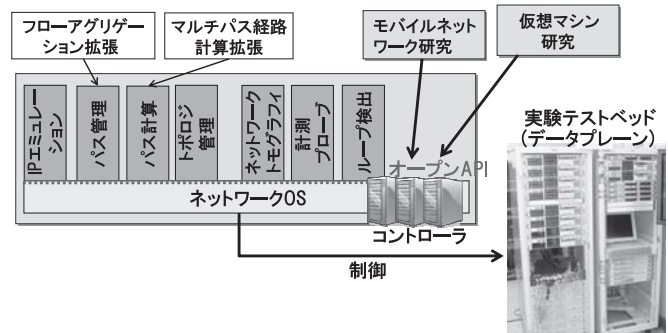


図6 テストベッド構成

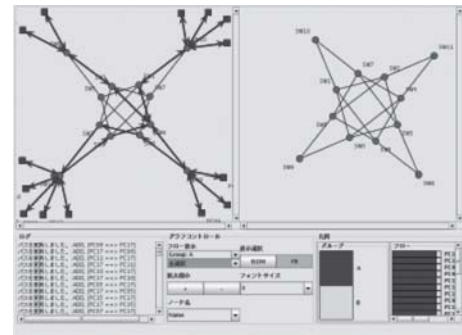


図7 仮想ネットワーク可視化

定・管理）、Path resolver（経路計算）、IP emulation（Ether/IPによるネットワークのエミュレーションを行う）などの制御モジュール群が実装され、以下のようなネットワーク機能の実験を行っています。

(1) 仮想ネットワーク可視化

図7 のように、仮想ネットワークのトポロジとその上を流れるフローを動的に可視化します。この図の例では単一物理ネットワーク上に2つの仮想ネットワークを構築し、それぞれを流れる各TCP/IPフローが3次ハイパーキューブネットワーク上でフローごとにロードバランスされています。

(2) フローベースメッシュネットワーク

フローベースでロードバランスされた大規模メッシュネットワークを実現するため、大規模マルチパス経路計算アルゴリズムを提案しPath managerモジュールの拡張として実装しました¹¹⁾。またスイッチの経路表を削減するためのフローアグリゲーション方式を提案し、Path managerモジュールの拡張として実装しました¹⁰⁾。

(3) 制御ネットワークの構築自動化

制御サーバとOpenFlowスイッチの間の制御ネットワークに対して、OpenFlowネイティブな構築方式を提案しました¹²⁾。

(4) ネットワークトモグラフィ

OpenFlowによる自由なフローパス設定を連携したアクティブプローブと、Topology discovery/Path managerと連携した品質劣化箇所推定方式を提案しました¹³⁾。

(5) フローループ検出

フロー単位でのパッシブ計測に基づくループ検出と、フロー単位でのループ除去方式を提案しました¹⁴⁾。

6. まとめ

本稿では、多様性のあるインターネットインフラを実現するために必要な技術要件として、オープン化、仮想化、モジュール化、そしてプログラマビリティについて述べ、それに基づくアーキテクチャを提案しました。そして、その技術的な核となる統合制御プレーンのためのネットワークOSを提案しました。本稿で提案したネットワークOSは、OpenFlowスイッチを含めた仮想インフラ全体を制御する統合制御プレーンのためのソフトウェアプラットフォームです。また、その実証のため本ネットワークOSやさまざまな制御モジュール群の試作評価を行い、その概要について報告しました。

参考文献

- 1) “AKARI,”
<http://akari-project.nict.go.jp/>.
- 2) NSF NeTS FIND Initiative,
<http://www.nets-find.net/>
- 3) Seventh Framework Programme (FP7),
<http://cordis.europa.eu/fp7/dc/index.cfm>
- 4) “GENI,”
<http://www.geni.net/>.
- 5) 下西英之, 石井秀治, 「統合制御プレーン向けネットワークOSの提案とそのOpenFlowコントローラへの適用」, 信学技報, NS2009-162, 2010.
- 6) Hideyuki Shimonishi and Shuji Ishi, “Virtualized network infrastructure using OpenFlow”, to be presented at BcN workshop, 2010.
- 7) N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peter-son, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol.38, no.2. 2008.
- 8) NOX | An OpenFlow Controller,
<http://noxrepo.org/wp/>
- 9) 加納敏行, 「次世代インターネット実現に向けた技術動向とNECの取り組み」, 電波新聞1月7日号特集, 2010.
- 10) 千葉靖伸, 下西英之, 篠原悠介, 「フローベースネットワークにおけるフローエントリ削減手法の提案とOpenFlowネットワークへの適用」, 信学技報, NS2009-163, 2010.
- 11) 篠原悠介, 千葉靖伸, 下西英之, 「データセンタネットワークにおける効率的マルチパスルーティングとそのOpenFlowネットワークへの適用」, 信学技報, NS2009-164, 2010.
- 12) 小出俊夫, 下西英之, 「OpenFlowネットワークにおける制御ネットワークの構築自動化に関する一検討」, 信学技報, NS2009-165, 2010.
- 13) 波多野洋一, 山崎康広, 下西英之, 「OpenFlowネットワーク上でのネットワーク品質劣化箇所推定における計測手法の効率化」, NS2009-166, 2010.
- 14) 伊藤暢彦, 山崎康広, 下西英之, 「メモリ使用量を削減したパッシブ型ループパケット検出方式とOpenFlow Networkへの適用」, NS2009-167, 2010.

執筆者プロフィール

下西 英之
システムプラットフォーム研究所
主任研究員