

# アプリケーション・サービスのためのネットワーク仮想化 Network Virtualization for Application and Service

新 麗

Ray ATARASHI

株式会社 IJ インノベーションインスティテュート IJ Innovation Institute Inc.

E-mail: ray@ijlab.net

概要: コンピュータサーバの仮想化技術は、コンピュータシステムを大きく進展させてきた。物理的なリソースをソフトウェアで隠蔽し、自由に生成したり分割したりする技術により、システムの柔軟性が飛躍的に向上したのである。さらに近年、ネットワーク技術の仮想化技術が注目を集めている。ネットワーク機器の持つ機能をプログラム可能としたり、ソフトウェア化したりすることで、コンピュータと同様の柔軟性を持つようになってきた。

Abstract: Virtualization technologies for Computer Server have developed the computer system. The technologies enhanced flexibility of the system hiding physical resource by software to create and split dynamically. Lately, network virtualization technologies are getting a lot more attention. Network technologies have gotten same flexibility as computer because network function became programmable and implementable by software.

## 1. はじめに

日本で商用インターネットサービスが開始されてから 20 年が経過した。平成 24 年度末のインターネット利用者数は、9,652 万人、人口普及率は 79.5%と発表されている [1]。年齢別に見ると 13-49 歳までは 9 割を越えているのに対し 50 代で 8 割、60 代で 7 割と年齢とともに下がっていくが、現在 9 割が使っている世代は年齢が上がっても使い続ける可能性が高いとすると、60 歳以上の利用率も上がっていくだろうと思われる。

インターネットの普及に伴い、アプリケーションの種類やサービスも多様化してきている。インターネットの利用目的は、メールの送受信、ホームページの閲覧だけでなく、商品・サービスの購入・取引も 5 割を越えている。利用者に見えるのはアプリケーションやサービスであり、インターネットを使っていることは意識されていないだろう。インターネットはアプリケーションに付属するものとしてとらえられている。

しかし、アプリケーションやサービスを提供する側にとっては、インターネットの機能や構成を意識して設計したりサービス構築したりする必要がある。インターネットサービスを提供する機器や PC サーバを組み合わせて、サービスに適したシステムを組み上げるという作業であり、専門的な知識を必要とする。IT 事業がサービスを提供していた時代は、専門知識を持つ SE が設計を行っていたが、多分野でサービス事業者が立ち上がるようになってくると、IT の専門知識を持つ SE が不足してきた。PC サーバやインターネットなどのインフラ部分を簡易に扱えるようにしたいという要求から生まれたのが、インフラ部分を仮想化したクラウド

と言える。

そのクラウド上では、多様なアプリケーションやサービスを提供するため、インターネットのシステムはますます巨大化、複雑化している。またクラウドサービスの特徴として、利用者側にハードウェアの更新が必要なくなったため、特定のサービスやアプリケーションを利用する期間が長くなったということが挙げられる。これまでは 5 年程度で老朽化するハードウェアの買い替えに合わせて設計を見直し新しく作りなおしていたシステムが、クラウドに移行したことで見直しの機会と動機が減ったのである。ハードウェアはやはり 5 年程度で老朽化するためクラウド事業者は入れ替えを行っているが、それは利用者側には見えない。問題は、ハードウェアを入れ替えた場合に、新旧の機器が完全に同じ機能を提供することが期待されることである。機能レベルでの互換性はあるものとなないものがあり、かなり高度な専門的知識と膨大な組み合わせの検証が必要とされるため、現実的とは言えなくなってきた。

クラウド技術の進展は、物理リソースを隠蔽して分割利用する仮想化技術が基盤となっている。物理的には 1 台のサーバ上に複数のシステムが稼働したり、複数のストレージをまとめて扱った上で領域に分割したりといったことが可能となった。これに対し、ネットワーク技術の仮想化技術は遅れていたが、ここ数年で急激に進んできた。中でも特に大きな注目を集めているのは、**Software Defined Network (SDN)** である。SDN はネットワークをプログラム可能にするという概念であり、この実現のためにさまざまな技術開発、および

標準化が進められている。本稿では、ネットワーク仮想化をとりまく環境として、SDN と SDN を実現する環境の一つとして重要視されている Openflow, CISCO の ACI などハードウェアベンダの対応と NFV (Network Function Virtualization)に付いて紹介する。

## 2. ネットワーク仮想化技術

ネットワークでは、古くから物理的な構成にとらわれず論理的に構成する技術を開発し使用している。Virtual Circuit など Virtual と名のつく技術はいくつも存在するし、システム構築やデータセンタネットワークなどで多用される VLAN は、仮想化技術のひとつである。P2P でよく利用される、下位ネットワークの物理トポロジを意識せず別のネットワークを構成するオーバーレイという技術もある。

このような仮想ネットワーク仮想化は、動的に変更することをあまり意識しておらず、インフラを運用するためにネットワーク技術者がネットワークのために構築するものであった。ところが、サーバやストレージの仮想化技術が進んだことで、ネットワークもシステムの一部として運用・設定を変更する必要が出てきたのである。これまではネットワーク管理者が主に手動で設計・設定していた仮想化の機能をシステムとして連携するために、設定インタフェースの仮想化・抽象化と、オープン化が急激に進んできた。

ネットワーク機器のインタフェース標準化としては NETCONF があり、クラウドシステムの構築および運用に利用されている。SDN の登場で NETCONF も見直されており、特にモデル部分に関しては関心が高まっている。

一方、SDN の導入・普及にあたって、各ハードウェアベンダによる様々な取り組みも始まっている。これまでのようにネットワークの基本的機能を高度化するのではなく、機能を利用するためのインタフェースを設計し API として公開するという方向性が主である。また、インタフェースを通じて機能を利用する場合、サービスやソフトウェアの開発者・利用者はベンダロックインを嫌うため、ハードウェアベンダもネットワークの機能そのものを仮想化し標準化する方向性へと動き始めた。本章では、これらの動向を概説する。

### 2.1 Software Defined Network (SDN)

SDN とは、ネットワークの構成、機能などをソフトウェアで操作するコンセプトである[2]。これまでのネットワークは、管理者が全体の設計図をもとに、個々の機器に対して設定を行うことで実現するのが一般的

であった。また、ネットワーク機器の管理とサーバの管理とは分かれていた。しかし、サーバの仮想化技術が進み、ネットワーク上に様々な仮想サーバが存在し、動的に生成・消滅するようになると、ネットワークも同様に構成変更・設定変更が必要となる。設定変更の頻度も上がったため、そのたびにネットワーク管理者に依頼して変更を待つという形態は、実用的とは言えなくなってきた。そこで提唱されたのが、ネットワークの構成・設定変更もソフトウェアで行う SDN という概念である。

SDN は特异的技術ではないため定義はさまざまであるが、共通する定義としては、データプレーンとコントロールプレーンの分離、全体を管理するコントロールソフトウェア、ネットワークがプログラム可能であること、が挙げられる。

SDN では、スイッチなどのネットワーク機器は、パケットをフォワーディングするデータプレーンであり、転送パスなどを制御するコントロールプレーンは別のソフトウェアが行う。そのソフトウェアが全体を管理する SDN コントローラである。スイッチとコントローラの操作プロトコルはオープンであり、プログラムすることで柔軟な機能を利用することが可能となる。図 1 に SDN の概念図を示す。

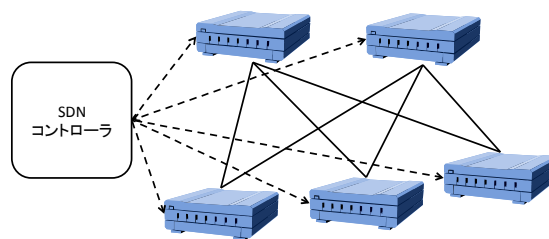


図 1. SDN の概念図

SDN においても一つ注目が集まっているのが、Northbound API である。SDN コントローラとスイッチなど各ネットワーク機器との間 Southbound API と呼ばれ、主に後述する Openflow でプロトコルが規定されつつある。また、既存の機器設定プロトコルでコントローラを実装する動きもある。それに対し Northbound API は、コントローラやその上位レイヤで提供される、ネットワーク構成や機能を制御するための API を指す。Southbound API はネットワーク機器の制御の API だが、

Northbound API は、アプリケーションソフトウェアからネットワークを制御するための API である。サービスやアプリケーションにとっては、ネットワークの仮想化は Northbound API の形で提供され、制御可能となると考えられる。詳細な検討や標準化はまだこれからではあるが、今後は期待されている。

## 2.2 Openflow

SDN を実現する代表的なものとして注目を集めているのが Openflow である[3]。OpenFlow は、ネットワークをプログラム可能とすることを目的としており、それを実現する OpenFlow Switch を設計・標準化している。OpenFlow Switch は、従来のハードウェアで提供されるフォワーディング機能としての Switch 部分をデータプレーンとし、コントロールプレーンとなる制御ソフトウェアを実装している。制御ソフトウェアは、さらにコンピュータ上などで動作するコントローラによって一括管理される。コントローラは、パケットのフローを管理して最適な経路を計算し、各 Switch の制御ソフトウェアを通して Switch のフローテーブルを設定する。従来の Switch はそれぞれ個々のフローしか見ておらず、全体の経路などは把握できないが、OpenFlow のコントローラは各 Switch の情報を集めて全体のフローを把握するため、最適経路を導き出すことができる。この機能を利用すると、例えば、あるユーザ間で急に大容量のデータ転送を始めたときに、そのフローを検出し、他のユーザに影響を与えないよう経路を変更したりすることができる。

OpenFlow は、大学のキャンパスネットワーク上に実験ネットワークを構築することを目的として開発された。集中管理を行うコントローラがボトルネックになるため、大規模なネットワークではスケールしないということは知られている。しかし、OpenFlow に対応する Switch ベンダが登場しはじめたことに伴い、コントローラの分散化、コントローラと Switch との通信量の削減、フローテーブルの圧縮など、大規模化に対応する研究も進んできた。現在は主にデータセンタのネットワークでの採用が期待されている。

現在の Switch は、ベンダによって仕様もインタフェースも異なるため、違うベンダの機器を一括で管理するのは困難である。OpenFlow は、OpenFlow 対応の Switch を標準として規定したため、対応している Switch であれば、同じインタフェースで管理できることがメリットである。現在、いくつかのベンダから Openflow 対応の機器が提供されている。

## 2.3 ACI (Application Centric Infrastructure)

ネットワーク機器ベンダの最大手である CISCO

Systems は、SDN への取り組みとして、Application centric Infrastructure (ACI) [4] を発表している。ACI は、ネットワーク、セキュリティに加え、仮想マシンやサーバ、ストレージが利用するネットワーク関連の設定を統合的に制御し、これをアプリケーションから利用できるようにする機能である。ネットワーク関連の設定を抽象化し、アプリケーションネットワークプロファイルとして提供する。アプリケーション担当者は、これを当てはめていけば設定なしに利用することができる。

Southbound API としては、Openflow ではなく、OpFlex という新たなプロトコルを提案し、IETF で標準化を進めていくとしている。すでに Internet-Draft[5]が提出されているが、CISCO だけでなく、IBM, Microsoft, Citrix など、アプリケーションや仮想サーバ提供者との共同提案となっている。プロトコルは、JSON か XML で交換される。まだ第 0 版でありこれからどう改訂されていくかは未知数であるが、大手ベンダからの提案は影響力が大きいため、今後の動向が注目されている。

## 2.4 OAN (Open Autonomic Networking)

日本唯一のルータベンダである、Alaxala Networks は、SDN が登場する前の 2007 年から、Open Autonomic Networking (OAN) という概念および API を提供している[6]。OAN では、メモリやディスク、CPU といったコンピュータリソースと同様にネットワークを扱うことを可能とすることを目的とした技術で、サーバやストレージ、ネットワークが一体となった自動運用を目指している。

ネットワーク、あるいはサーバの運用者が環境に合わせてコントローラを実装できるよう、API とともに Java ベースの SDK である AX-ON-API-SDK が提供されている。Southbound API としては NETCONF をベースとしている。

## 2.5 Network Function Virtualization (NFV)

通信事業者も、ネットワーク仮想化に向けて動き出している。Network Function Virtualization (NFV) は、ハードウェアで実現されていた通信事業者のネットワーク機能をソフトウェア化し、汎用サーバの仮想化された基盤上に集約していく仕組みである。通信事業者から提案され、ETSI (European Telecommunications Standard Institute) で標準化が進められている。NFV はモバイル通信においても注目されており、携帯コア網である EPC (Evolved Packet Core)、サービス系のシステムである IMS (IP Multimedia Subsystem)、無線アクセス網における集中化したベースバンドユニット BBU の仮想化なども検討されている。

SDN とは補完関係にあるとされており、SDN はネットワークを抽象化することでネットワークの設定変更を容易にし、NFV は設備、運用コストを下げることを可能とする。

### 3. サービス定義によるネットワーク

著者らは、ベンダによるネットワークの仕様の差異を吸収して一括管理できるよう、データベースを構築し、NETCONF を利用してネットワーク機器を制御する方法を提案してきた。しかし、データセンタ等に適用するには、ネットワークだけでなくサーバやストレージ等の管理も必要であり、ネットワーク機器だけの管理では不足することがわかってきた。そこでこのたび、データセンタ等で提供する機器の組み合わせをサービスと定義し、サービスを規定すると自動で構成するインフラ、Service Defined Infrastructure (SvDI)を提案している[7][8][9]。

SvDI は、仮想サーバ、ストレージ、ネットワーク機器をリソースとして一括して扱う概念である。リソース割り当て要求に対して、仮想サーバ、ストレージを割り当て、それらを接続するネットワーク機器を選択し接続するための設定を行う。サーバのみ、ストレージのみを増減する場合にも同様に割り当てと設定を行う。サービスには種々あるが、例えばストレージとサーバとのアクセスが多いサービスであれば当該の通信がスムーズに行くように考慮し、サーバとインターネットとの通信量が多いサービスは、外部接続を強化するなどの対応を行う。

SvDI の目的の一つは、数年ごとにハードウェアを交換しながらも、サービスを維持しつづけることにある。サービスを維持するためには、ハードウェアとその機能の抽象化が必要であり、現在、モデル化を進めている。

### 4. おわりに

サーバから始まった仮想化技術が、クラウドやデータセンタでの運用の要求を受けて、ネットワークへと広がってきた。サーバとの協調も進んでおり、データセンタの運用はこれから大きく変わっていくと思われる。データセンタの機能が高度化すると、アプリケーションやサービスへも影響があるだろう。利用者にとっては、サーバやネットワークを意識せずにアプリケーションを利用できるのが理想である。IT 技術がますます使いやすくなり、コミュニティの活性化につながる方向になるよう期待している。

## 文 献

- [1] 総務省 情報通信白書 平成 25 年度版,
- [2] McKeown, Nick. "Software-defined networking." INFOCOM Keynote talk, 2009.
- [3] N. McKeown et al., "OpenFlow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review, Volume 38 Issue 2, Pages 69-74, April 2008.
- [4] Cisco Application Centric Infrastructure, <http://www.2mul.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/index.html>
- [5] M. Smith et al, "Opflex Control Protocol", IETF, <http://datatracker.ietf.org/doc/draft-smith-opflex/>
- [6] Alaxala Networks, Open Autonomic Networking, <https://www.alaxala.com/jp/products/oan/>
- [7] 寺本泰大, 岡部寿男, 新麗, "トポロジーデータベースを利用したネットワークの自動設定変更", 電子情報通信学会技術報告[インターネットアーキテクチャ], pp.21-26, IA2011-58, 2012 年 1 月.
- [8] 新善文, 新麗, 岡部寿男, "Service Defined Infrastructure(SvDI)の実装方法と応用の検討", 電子情報通信学会技術報告[インターネットアーキテクチャ], pp.167-172, IA2011-95, 2012 年 3 月.
- [9] 新麗, 金田克己, 加藤雅彦, 須賀祐治, 岡部寿男, "IaaS/HaaS への Service Defined Infrastructure (SvDI)適用の検討, 電子情報通信学会技術報告[インターネットアーキテクチャ], pp.37-42, IA2012-18, 2012 年 9 月.