



SEMINAR REPORT

最新の OpenFlow/SDN 技術とこれから期待される応用



本日は OpenFlow、それから SDN (Software Defined Networking) についてお話ししたいと思います。どういう時代背景で始まって、どういう思いで始めたのか、最新の標準化動向についてもせっかくの機会なのでお話ししたいと思います。また、OpenFlow の特徴の一つであるオープンイノベーションが、どのようにして行われてきたのか、実際のデータセンターとクラウドシステム、キャリアネットワークに向けてどのような応用が考えられるのか、といったこともご紹介したいと思います。

OpenFlow/SDN 登場の背景

2006 年頃の時代背景として、そろそろインターネットの限界を何となく世の中の人達が意識し始めた時代です。

図 1 に示すように、1970 年代からインターネットは始まっています。例えば、RFC1 は 1969 年です。最初のインターネット通信、ARPAnet (Advanced Research Projects Agency Network) 上の通信が 1969 年 10 月 29 日 10 時半に行われました。これは失敗でした。確か、遠隔ログインしようと"login"と打とうとして、"lo"と打ったところでハングアップしてしまい、世界初のインターネット通信は 2 文字で終わってしまったというのが、私が関係者から聞いた話です。TCP の原著が 1970 年代に出され、その後 IP の RFC が出されて、本格的にインターネットが始まりました。

1970 年代に生まれた技術で他に何があるのかネットで調べてみると、リアルタイムプログラミング言語 HAL/S があります。これはスペースシャトルの機体制御に使われたコンピュータ言

語らしく、インターネットと違ってこの技術には終わりがあり、一昨年スペースシャトルの最後の打ち上げでこの技術の寿命は尽きました。それに比べて、インターネットが今でも十分現役で使われているのは、よほど最初の設計が良かったことが当然あると思いますが、インターネットで使われている対象に大きな変化があったことも関係あります。

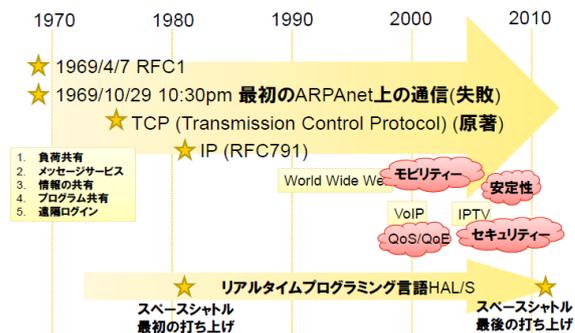


図 1 インターネットはいま何歳?

図 1 の左端の記述にあるのは、最初の ARPAnet の設計仕様に書かれていたプロジェクトの目的で、主にコンピュータ間の情報交換を目的としています。ここから始まったインターネットが、例えば、メディアや World Wide Web などに使われるようになるにつれて、赤の雲で示すように、モビリティは、安定性は、セキュリティは、QoS はどうかと、非常に疑問が持たれるようになってきました。

こういったところから、インターネットのインフラを考え直すという気運が、2005~2006 年当時の人達の心に芽生えてきました。なぜ米国政府が 400 億円もの予算をかけて GENI (Global Environment for Network Innovations) プロジェクトを始めたのかは、こういった背景があるからだ当時話されました。

当時はインターネットが大いに花盛りで、イノベーションがとても活発に行われていました。インターネットでのイノベーションというものの、残念ながらインフラには光はあたっておらず、ほとんどの学生はユーチューブ、フェイスブック、グーグルなどのインターネットを使った新しいサービスに関心を向けていて、そういったところではとても盛んにイノベーション

が行われていました。コンピュータサイエンスを志向する学生自体もますます減ってきて、その中でネットワークを志向する学生も同様に減ってきているのは非常に問題視されていました。しかも、インターネットを支えるインフラは多くの問題を抱えており、今後ともイノベーションを必要としています。こういった状況に非常に危機感を持って、どうしたらインフラにイノベーションを取り戻せるのか、学生を引きつけられるのかというのを非常に危惧しておりました。

インターネットを用いたサービスでは、例えば、大学生などがフェイスブックなどのような新しいサービスを始めたいと思いついたとき、アマゾンのサーバを借りてくれば、当面のプログラミングスキルさえあれば、誰でもサービスを始めて、そこからイノベーションに成功すれば、一夜にして大金持ちになると、非常に魅力的な領域でした。それに比べて、インターネットのインフラそのものは、大手のベンダーが莫大な開発費をかけて、例えば、ルータや交換機を開発して、それをキャリアに納めて5年10年と運用するとなると、学生がどんなに面白いアイデアを考えたとしても、イノベーションが実際に世の中で花開くまでに学生が学生でいられなくなるほどの時間を費やしてしまいます。

このような状況に非常に危機感をもって、それではどうやってインフラのイノベーションを持続的に回したらよいか、インフラを再発明しようかと、先ほどのGENIやその他色々な動きが始まりました。注目を浴びていなかったところに再び光をあてて、インターネットのインフラに対して人々の興味を集めて、イノベーションを活発にしていこうという機運です。

図2が2006年当時の世の中の状況で、日本、欧州、米国で色々なプロジェクトが走っていました。一つ注目すべきことは、図の上半分は通常のリサーチファンドで、色々なリサーチに対してファンドがありましたが、下半分はテストベットに対するファンドで、国家予算による資金投入が大きな割合を示していたのが特徴でした。

2020年頃に求められる新しいネットワーク(NW)アーキテクチャを研究開発

	米国	欧州	日本
研究開発プロジェクト	FIND ・将来のインターネットのアーキテクチャの確立を目指すNSFの研究ファンドプログラム	FP7 ・欧州域内の大学や企業の技術力や競争力確保を目的とした研究開発への助成プログラム	AKARI ・新しいNWアーキテクチャの確立と、それに基づくNW設計、実証実験等を通じて新世代NWの実現を目指す
テストベット実証プロジェクト	GENI ・独自の多様なアーキテクチャに対応するためのグローバルな実験設備構築を目指す	GEANT ・欧州全域をカバーする研究開発NW	JGN2plus ・産学官連携によるNW関連技術の研究開発や実証実験等に活用するテストベットNW

出典: NICT「新世代ネットワークに関する取組について」をもとに作成

図2 Future Internet に向けた日欧米の取り組み(当時)

これはどういうことかという、例えば、インターネットの仕組みで新しい事業を考え着的なとき、プラットフォームがベンダーの装置にインプリされないと世の中に出せないのであれば、イノベーションしようと思っても、モチベーションがわきづらい面があります。そうすると、考えたものが実際に装置として動作し、それがインフラとして運用されるようなテストベットが必要になります。色々な新しい考えを持っている人が、テストベットによる運用を始めて、そこで通信サービスとして成り立つようになれば、それは大きなインセンティブになると思わ

れます。テストベットに対して大きなファンドが付くようになったのが、欧米では2006年以降、日本では2007~2008年以降です。JGN2plus (Japan Gigabit Network 2 plus) はそれ以前からありましたが、JGN2plus 上でのOpenFlowのテストベットが始まったのは2009年以降です。このあたりが当社の手掛けてきたところなんです。

新世代ネットワークに関する2006年~2008年の状況があって、テストベットを作ろうという機運が各国で盛り上がってきた中で、その内の実用化技術の一つとしてOpenFlowが生まれました。当時、OpenFlow以外にも色々なオプションがありましたが、最終的にこのOpenFlow技術が残ったわけで、テストベットが実際に動き始めたところから、以降注目を浴びた技術です。

OpenFlowとは何かを図3にて説明します。一般にネットワーク機器は二つの機能要素から成り、一つは、単純にデータを右から左側へ流すだけの、純粋にデータパケットを受け渡しするフォワーディング機能であり、ハードウェアの進化に伴ってますます高速になっていく部分です。もう一つは、頭脳に相当するところで、例えば、隣接にどのようなルータが存在している、どの経路でパケットを流せば、どの様にパケットが届くだろうという、ネットワークのインテリジェンスを司る非常に重要な部分です。

データプレーンと制御プレーンを分離してオープンなプログラマビリティを実現

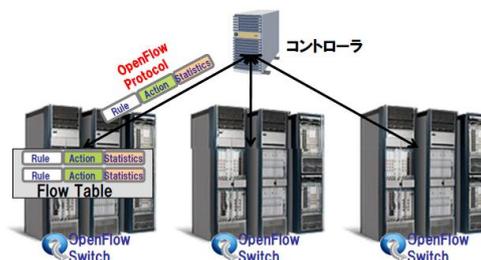


図3 OpenFlow とは？

ネットワークで新しいことを導入するとき重要となるのは、データをただ単に右から左に流すフォワーディングの方ではなく、ネットワークをどの様に制御して、どの様にパケットを流すかというインテリジェント部分の方です。前者は汎用的な機能としてハードウェアによる高速化を進めればよい領域です。後者のインテリジェント部分は、前者から切り離して、コントローラと呼ばれる汎用サーバを用いるというのがOpenFlowの基本的な考え方です。

そうすると、従来はネットワークにインテリジェンスを持たせようとする、両者の機能が一つの機器に一体化されていると、そこに新しい機能を実装するのは容易ではありませんが、インテリジェント部分を切り離して誰でも触れるようになって、その間にオープンな標準プロトコルを定めると、その上のインテリジェンスは誰でも設計できるようになると考えられます。こういったものをテストインフラとして例えばGENIなどでデプロイメントして、コントローラ上のプログラミングを誰にでも許すようにすれば、誰でもイノベーションに参加し、新しいことを実際に運用されているインフラ上で直ちに試すことができるようになります。これは大変面白い試みだということですが、

OpenFlow のスタート地点なのです。

ネットワークの制御サイドを切り離して、コントローラ上に置いたとすると、当然ネットワークの制御だけでなく、例えば、バーチャルマシン(VM)の制御とか、サービスの制御とか、それこそネームサーバやセキュリティなどの色々なインフラを制御する機能などが、同じようにサーバ上にインプリされます。今まではそういった IT インフラ側のインテリジェンスとネットワークは完全に切り離されていましたが、図4に示すように、ネットワークのインテリジェンスだけを上に切り出して、IT インフラ側のインテリジェンスと一緒にしたとき、従来ネットワークだけでは提供できなかった新しいサービスが可能になると言われています。そして、ネットワーク技術のイノベーションだけでなく、ネットワーク技術を含んだインフライノベーションが、このコントローラ側の役目として存在することになります。

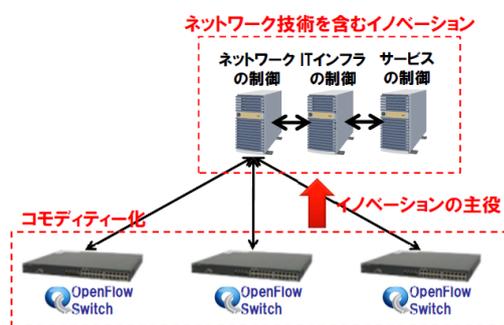


図4 OpenFlow とは? (つづき)

OpenFlow の活用方法としては、当時、色々考えて幾つか実践しました。図5に例を示します。

一つは、キャンパスネットワークです。学生が授業や研究室で何かネットワークの新しい仕組みを考えようとプロジェクトを始めたとき、自分達が考えたネットワークの仕組みを OpenFlow を使ってキャンパスネットワークで実践してみます。例えば、新しいメールデータを流す仕組みを作ったとき、実際に自分達がキャンパスで使っているメールを配信してみます。すると、自分で思ったことが実際にキャンパスネットワークで動いて、それを皆で使ってみて面白いというのは勿論インセンティブになるのですが、研究室のテストネットワークで動かすだけでなく、実際に実ユーザーを募ることで、更に新しい課題を発見し、次のイノベーションの種になるということです。

次に考えたのはデータセンターネットワークです。この図5では、サーバは左から右に動いて行っていますが、例えば、夜中のデマンドが少ないときにサーバの数を減らして、デマンドを1台のサーバに集めて、残りのサーバをシャットダウンして消費電力を下げようというときに、単純にサーバ、仮想マシンを移動するだけでなく、そこに張っていたネットワークパスも同時に切り替えます。ネットワークとサーバ制御の仕組みを OpenFlow コントローラで一体化すれば、こういったことが自由にできるようになるのではないかと、そうすると、これは新しいデータセンターの運用形態として使えるのではないかと、いった議論もありました。

あとは、企業ネットワークの話です。OpenFlow コントローラがネットワークのコントロールを司るのであれば、例えば、セキュリティアプライアンスや従業員の認証システムなどのサ

ーバの認証システムとネットワークアクセス制御をきちんと繋ぎ合わせることによって、よりセキュアなプライベートネットワークが構築できます。

他には、キャリア網をモバイルのバックフォールにするとか、有線系のキャリアのアクセス系にするとか、サービスネットワークにするとか、こういった議論を OpenFlow の活用事例として検討しています。

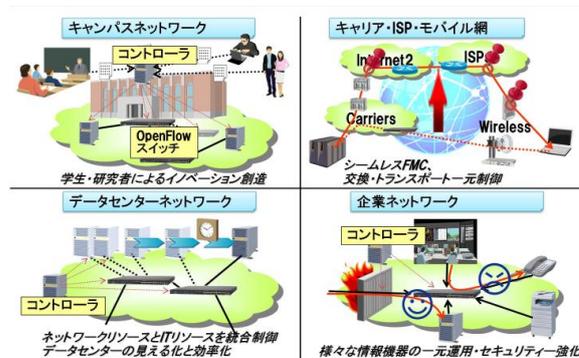


図5 OpenFlow を活用した研究開発

そして、最初はフューチャーインターネットという呼び方で始まったものが（日本では新世代ネットワークという呼び方でしたが）、OpenFlow というのが生まれて、最近はその総称として、ソフトウェア・ディファインド・ネットワーク (SDN) と呼ぶようになってきています (図6)。

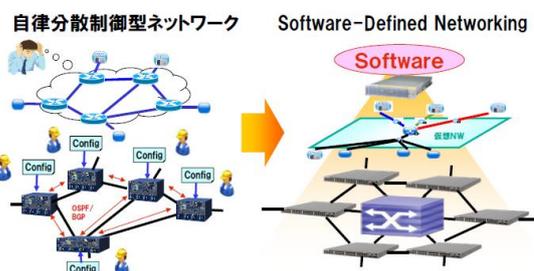


図6 そして、SDN (Software-Defined Networking)

それでは何故、OpenFlow がここまでうまくいったのでしょうか？ 図7に示すように、幾つかの要因が重なったような気がします。

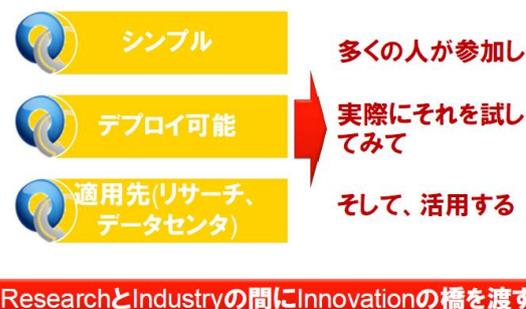


図7 そして、OpenFlow

一つは、非常にシンプルに物事を捉えようとしたことです。データパスとコントローラ部分を切り離すという考え自体は、

OpenFlow が最初に始めたのではなく、色々なところで昔からありました。

ネットワーク分野の周辺にいる若い人達やネットワークが本業ではない人達が、色々なイノベーションをネットワークの世界で起こすようになるためには、ネットワークの世界が閉じているのではなく、色々な人が簡単にこの世界に飛び込んで来られるように、とにかく物事がシンプルになるようにしました。

仕様をシンプルにするだけでなく、動作ソフトのオープンソースコードを最初から提供して、誰でもコードを書けば始められるところから始めました。そうしたら、まず、大学の学生達が使い始めて、それが徐々に広がって行って、そしてクリエイタルマスを超えました。

もう一つ、デプロイメント可能ということをとっても重視しました。これはどういうことかと言うと、例えば、新しい方式を考えて、それを実験室のサーバに実装して動かしてみても、実験室のサーバ上のソフトでは、当然ながら実用には耐えられない。実用化しようとする、新しく交換機を作るなどハードウェアから始めたのでは、今度はコストや時間がかかってしまいます。そうではなく、考えついたアイデアがストレートに、それも実環境で使えるような速度のハードウェア上で速やかに実現しないことには、イノベーションがテイクオフしません。そこで、既存のハードウェアをそのまま、ファームウェアを書き換えるだけでこのプロトコルが実現できるように、非常に注意深く仕様をデザインしました。新しくハードウェアを作ったり新たにチップを開発したりせずに、今あるハードウェアのファームウェアを少し書き換えるだけで、実用可能なスペックを有した OpenFlow のハードウェアができることを極めて重視しました。

そういうところをきちんと考えていたから、単に実験室の道具に終わらず、実用にまで素早くたどり着けたということで、最初の頃のベンダーである当社と大学の学生達がこれだけうまくやってこられたのは、当社がベンダーとしてハードウェアを提供したその上に、彼らがソフトウェアを乗せていくという両者の win-win の関係でスタートできたからです。

三つ目は、適用領域が注意深く選択されていたということだと思います。OpenFlow では最初の適用先としてアカデミックに持っていきました。色々なオープンネットファンドがあり、その関係者達がイノベーションのためにプラットフォームを必要としていて、そこに話を持って行って、GENI などの政府系資金を使ったプロジェクトが広まっていきました。

その次に OpenFlow の適用先としたのがデータセンターです。クラウドが流行ってきて、データセンターをより効率的に使いたいところに、OpenFlow がうまく使えるのではないかと考えられました。こういった新しい技術が即使えるような状況を見定めて、常に OpenFlow をターゲットにしていきました。

OpenFlow というのは、技術的に誰もとも思いつかないようなことを思いついて実行したというよりも、非常にシンプルな技術を種として、世の中の状況を見定め、周りの人達を巻き込んで、どうすれば世の中に広がっていくかをしっかり考えて、テクノロジーマーケティングしてきたところが、とてもうまく回った事例ではないかと思っています。多くの人が参加し、実験室ではなく実際に運用してみて、そして、それがキャンパスネットワークやデータセンターで活用されるようになりました。リサーチからインダストリーにどうすれば繋がるかをしっかり

と考えながら、関係者を集めて働きかけたことで、最初はリサーチ環境から始まったものが、最終的にはインダストリーへと繋がったということです。リサーチ・インダストリー間のイノベーションの橋渡しがうまくいったのが、今日 OpenFlow がこれだけ受け入れられている理由ではないかと思っています。

OpenFlow 関連標準化動向

現在では、OpenFlow の仕様は、ONF (Open Networking Foundation) という標準化団体で議論されています (図 8)。

まず、2007 年にスタンフォード大学、NEC、NTT ドコモ、ドイツテレコムなど 5 社によるクリーン・スレート・プログラムという活動を始めました。これが OpenFlow の仕様の原点です。その後、2009 年にクリーン・スレート・ラボラトリー活動、そして発展的に解消して、最初の仕様書が作成されました。その後、2011 年 3 月に ONF という標準化団体を正式に発足しました。この間、4 年間のタイムラグがありますが、これは既存の標準化団体に話を持っていくのではなく、自ら新たな標準化団体を立ち上げるべく、4 年の期間をかけて準備したということです。

- 2007年: Stanford大学 Clean Slate Program活動開始
- 2009年: Stanford大学 Clean Slate Laboratory活動開始
- OpenFlow Consortiumでの OpenFlow spec(v0.8.9, v1.0), Trial活動
- <http://www.openflow.org>



- 2011年3月: Open Networking Foundation標準化団体発足

Industry
標準化活動



- 2012年5月: Open Networking Research Center(ONRC)発足

Industry
Open Source活動

Open Networking Research Center
at Stanford University

図 8 OpenFlow / SDN の活動の経緯

オープンイノベーションに向けた取り組み

そもそも OpenFlow, SDN を始めた理由は、ネットワークに対してイノベーションが起こるような環境を作ることが目的でした。実際、どの様なことを行ってきたかという、最初にスタンダード大学と一緒に OpenFlow を立ち上げたときに、当社が持っているイーサネットスイッチのハードウェアを改造して、OpenFlow の機能が動作するようにしました。

プロトタイプを作った後に最初に行ったことは、それを様々な研究機関、特に米国の大学に提供し評価をして頂きました (図 9)。

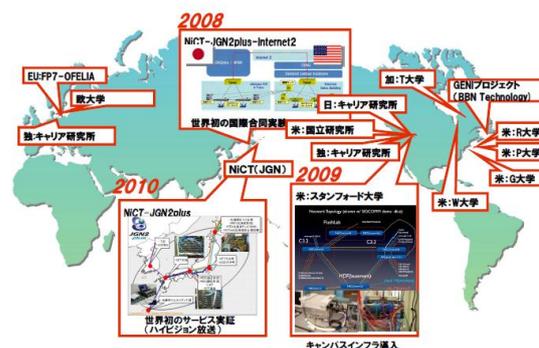


図 9 スイッチ試作機開発と研究機関への提供

図 10 は、当社とスタンフォード大などが一緒に行った第 3 回 GENI Engineering Conference での実証実験です。スタンフォード大に OpenFlow のネットワークを構築して、日本側は NICT の JGN2plus に OpenFlow のスイッチを配置して、太平洋を跨いだインターネット通信で JGN2plus とスタンフォード大の回線とを接続して、OpenFlow のネットワークによる仮想マシンの移動などを評価しました。

- 2008/10/28-30の米国GENI Conf 3rd: OpenFlowスイッチ動態デモ:
 - JGN2plus – Stanford大学内を相互接続してOpenFlowデモ
 - (1) Flow Dragging (自由なフローパスの動的設定/変更)
 - (2) VM / Session 動的ハンドオーバー (Latencyの短いサーバへVMが移動)
 - (3) 太平洋横断 VM Migration

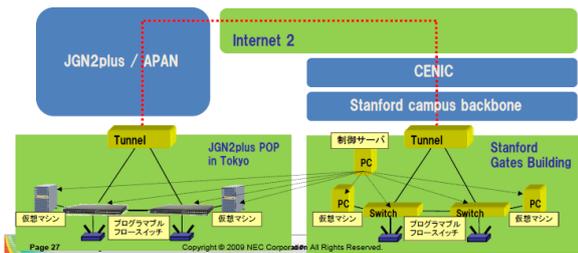


図 10 OpenFlow Switch 実証実験(GEC3rd)

もう一つ我々が進めてきたことは、コントローラの開発を誰でも簡単にできるようにしたいということでした。誰かが新しいネットワークの仕組みを考えて、それをコントローラに組み込みたいと考えたとき、いとも簡単に実現できるようにコントローラ開発のためのフレームワーク (Trema) を開発して、オープンソースとして公開しています (図 11)。

**Trema ≠ OpenFlow Controller
= Full-Stack OpenFlow Framework in Ruby and C**

- Tremaはコントローラではなく、新しいコントローラを簡単に開発するための"フレームワーク"です
 - OpenFlow/SDNのソフトウェア生産性を向上します
 - だれでも簡単にコントローラを開発をできるようにします
 - 便利なコントローラのサンプルも公開しています
- NECの研究所が開発 → オープンソース化(GPLv2)

図 11 コントローラのオープンソース活動

実際、このフレームワークを用いてベンチャー企業の人々がセキュリティアプライアンスを作るなど、色々な形でエコシステムが広がりつつあります。本を書いたり、色々なセミナーを開催したり、大学で OpenFlow/Trema の授業をしたり、学生に実際に OpenFlow のコードを演習として書いてもらったりと、色々なオープンコミュニティの活動を行っています (図 12)。

図 12 コミュニティ活動

データセンター・クラウドシステムソリューション

データセンターのクラウドソリューションとしては、どういったことを考えているかと言いますと、OpenFlow のユースケースの一つとして述べたことですが、データセンターを簡単に構築することです (図 13,14)。データセンターのインフラを構築するのを、IT 資源プールを作り、OpenFlow のネットワーク資源プールを作った後、プロビジョニングシステムによる仮想ネットワークと仮想サーバを構築することで、速やかにシステム構築できるようにすることです。

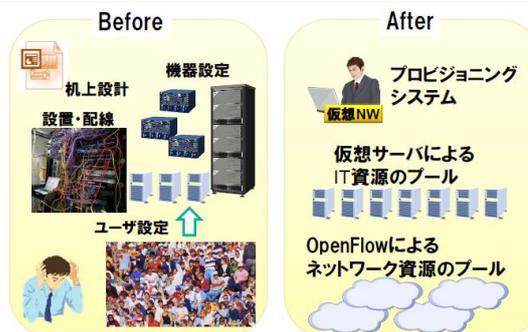


図 13 IT・NW 統合管理によるシステム構築迅速化(1/2)

- システム構築・運用の自動化により、コスト削減およびサービス導入迅速化を実現
 - サービス構成や需要に応じたIT・NW機器のプロビジョニング

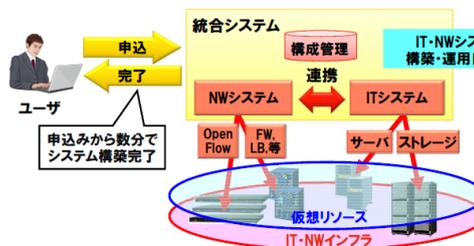


図 14 IT・NW 統合管理によるシステム構築迅速化(2/2)

従来はこういったデータセンターのインフラを構築するとき、IT ネットワークインフラがあって、サーバは何台必要で、それを繋ぐための仮想ネットワークは、LAN は、ファイアウォールはと設計していくと、以前は大概2週間ぐらい要していました。

それに対して、ユーザーが申込の後、仮想サーバプールして、VLAN 設計、ネットワーク仮想化設計して、2 週間かかっていたものを 5 分で提供することが一つの例です。

キャリアネットワーク向けソリューション

今までの OpenFlow の話は、最初に大学やリサーチ機関への導入、続いてデータセンターや企業ネットワークへの導入についてでしたが、最新動向としては、キャリア向けに OpenFlow 導入の画策が始まっています。

SDN のキャリアネットワークにおける活用例として有力なのが ETSI (European Telecommunications Standards Institute) で議論されている NFV (Network Function Virtualization) です (図 15)。NFV そのものは SDN がなくとも実現できますが、NFV を実現するネットワークインフラとして SDN を活用することで相乗効果を得られるということが議論されています。

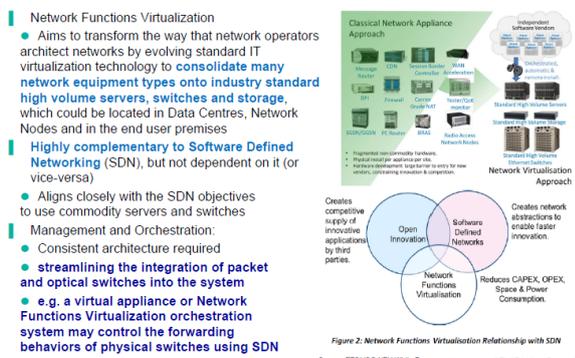


図 15 ETSI NFV (Network Function Virtualization)標準化

図 16 はキャリア側の ETSI ネットワークの例を示しています。例えば、ブロードバンドアクセスサーバとか、ルータとか、セッションボーダーコントローラとかのアプライアンス機器を次々と導入して行くと、維持費用はかかるし、サービス内容も変わって行く中で専用機器は導入しづらいです。この様なアプライアンス機器を全て仮想マシン化して、ソフトウェアで対応しようというのが NFV でやろうとしていることです。極論すると、今までキャリアの終端局に設置されていたルータなどの装置群を、データセンターのようにサーバをたくさん並べて LAN スイッチで結ぶ。後は色々なサービスをサーバのソフトウェアによって実現してしまうことを検討しています。

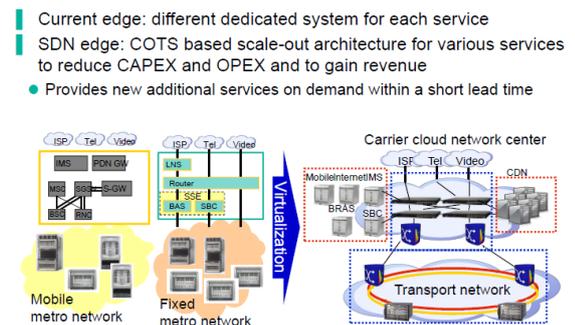


図 16 Network Function Virtualization(1/2)

図 17 は NFV をキャリアのエッジネットワークに適用した例です。例えば、赤線で示すようなサービスがアクセスから入って、BRAS (Broadband Access Server)、NAT (Network Address Translation)、IPS (Intrusion Prevention System) を経て、ルータからインターネットへと抜けていくという機器構成のとき、この様に個々に機器を置くのではなく、全てサーバ上の仮想マシン (VM) に置き換えて、VM でこれらのネットワークの処理を行うようにする。こうすることで、サービス需要が変わったときでも、VM を組み替えるだけで機器構成が変更られます。長期的な投資の面からも、色々なサービスごとに選んで機器を設置するのではなく、とりあえずサーバをたくさん並べておいて、必要に応じて VM で対応すれば、提供サービスの変化に対応してキャリアのウェブサービスが構築できると想定しております。

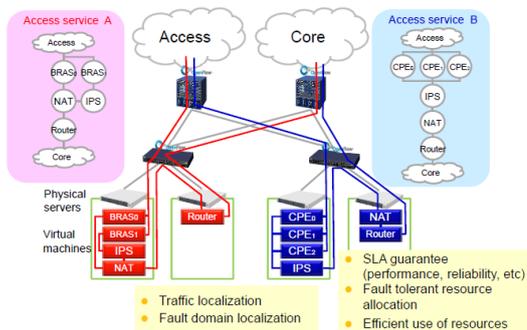


図 17 Network Function Virtualization(2/2)

最後に

そもそも OpenFlow や SDN というのは、どういうことを意図していたのでしょうか？ 産業構造という観点で捉えると (図 18)、昔はアプリケーション、専用 OS、専用ハードウェアというメインフレームの時代でした。それがいつの間にかハードウェアは PC などのコモディティになって、OS はマイクロソフトやリナックスに標準化されて、OS 上で色々なアプリケーションが動作するという、まさにコンピュータ世界のオープン化によって、このような変革がもたらされました。

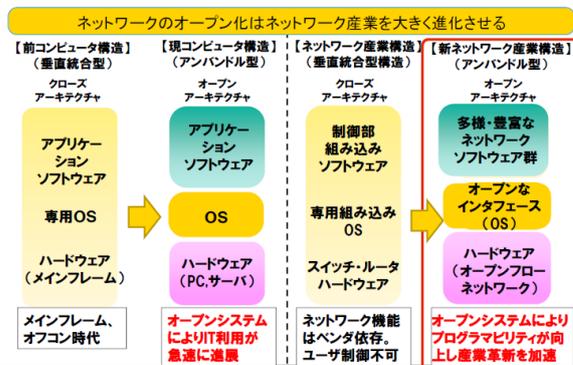


図 18 OpenFlow が変えるネットワーク産業構造

それと全く同じことが、ネットワークの機器においても起こるのではないのでしょうか？ スイッチやルータでは専用の組み

込みOSが動作し、その上で例えばBCP (Business Continuity Plan) 対策を支えるソフトウェアが動作し、これがベンダーのビジネスを支えているわけですが、これがOpenFlow ネットワークになれば、コモディティのハードウェアに内蔵されているのはフォワーディングチップだけで、そこには制御ソフトは動作していません。その上にはオープンなインターフェイス、オープンなオペレーションシステム、ネットワーク機器用のオペレーションシステムが存在し、さらにその上にサービスソフトウェアとして、色々なネットワークプロトコルが動作するのであると思います。

コモディティ上で動作する世界になると、ハードウェアは共用化されて、その上で動作するソフトウェアの違いで、先ほどのデータセンターの例では制御プログラムが、キャリアエッジネットワークの例ではエッジネットワークソフトウェアが適用されるということになります。そうすると、コモディティのハードウェアを製造する人、ユースケースごとに色々なアプリケーションソフトを作る人と、立場が分かれていくのではないかと、OpenFlow は意味しているように思います。

本講演録は、平成 25 年 6 月 14 日に開催されました、SCAT主催の「第 90 回テレコム技術情報セミナー」、テーマ「最新のOpenFlow/SDN 技術とこれから期待される応用」の講演要旨です。

*掲載の記事・写真・イラストなど、すべてのコンテンツの無断複写・転載・公衆送信等を禁じます。