

統合管理型ネットワーク仮想化基盤技術

山田一久 中尾彰宏 金田 泰 才田好則 雨宮宏一郎

進化型ネットワーク仮想化を実現する統合管理型ネットワーク仮想化技術について、従来のネットワーク仮想化基盤からの新たな追加拡張機能の研究開発成果について報告する。ネットワーク仮想化基盤の要件である資源抽象化、資源分離独立性、資源融通性、プログラム可能性、認証性を満たしながら、進化可能なアーキテクチャを提案するとともに、エッジ・ネットワークや端末の仮想化への拡張を行い、テストベッドへの展開を行った。

1 まえがき

現在の通信基盤が抱える様々な課題の全てを解決する革新的な設計思想を持つ情報通信基盤の構築には、進化型ネットワーク仮想化 (Advanced Network Virtualization) の考え方が必要となる。情報通信基盤は広義には、データを伝達する通信のための資源を有する「リンク」とプロトコルを解釈しリンクを選択するプログラムを実行する計算資源やストレージ資源を有する「ノード」を基本要素として構成されるが、進化型ネットワーク仮想化では、この広義のネットワーク全体を仮想化して利用する。具体的には、(1) 資源の抽象化 (Abstraction)、(2) 資源の独立分離性 (Isolation)、(3) 資源の融通性 (Elasticity)、(4) プログラム可能性 (Programmability)、(5) 認証性 (Authentication) の5つの要件を満たしながら、ネットワークを自由な発想で白紙から創造・設計することを可能とし、さらにそれらを複数同時に収容する基盤 (メタアーキテクチャ) を実現する新世代の重要情

報通信基盤を構築する。本稿では、このような進化型ネットワーク仮想化を実現するための統合管理型ネットワーク仮想化基盤技術の研究開発について述べる。

2 ネットワーク仮想化基盤の概要

我々が研究開発を行ってきたネットワーク仮想化基盤¹⁾の全体構成を図1に示す。ネットワーク仮想化基盤は仮想ネットワーク (スライス) を実現する仮想化ノード (VNode)、スライスと利用者 (ユーザ) との接続を提供するアクセスゲートウェイ (AGW)、ネットワーク仮想化基盤の資源を管理しスライス開発者 (デベロッパー) からのスライス要求からスライスを生成するネットワーク仮想化基盤管理システムから構成される。仮想化ノードはネットワーク機能をソフトウェアで実現する仮想ノードを提供するプログラマと、仮想ノード間を接続する仮想リンクを提供するリダイレクタから構成される。さらにエッジ・ネットワークや端末も仮想化し、スライスを提供する。

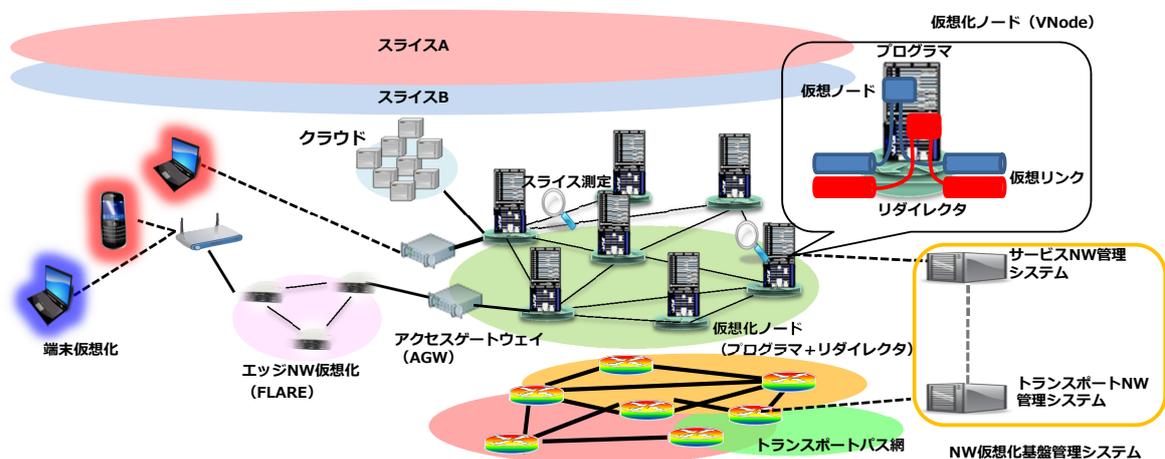


図1 ネットワーク仮想化基盤の全体像

3 研究開発内容

本研究開発では従来のネットワーク仮想化基盤に機能拡張や新規機能追加をすることにより、より高機能かつ高性能なネットワーク仮想化基盤を提供するとともに、エッジ・ネットワークや端末、他の仮想化基盤へのスライス拡張を実現している。以下に具体的な研究開発内容について述べる。

3.1 進化可能なネットワーク仮想化基盤アーキテクチャ

ネットワーク資源のスライシングにより、複数の独立に柔軟にプログラムが可能で、将来の要素技術と共に進化が可能なネットワーク仮想化基盤のアーキテクチャの研究開発を行ってきた。持続的にシステムが進化するには、リンク技術やプロセッシング技術、及びストレージ技術がそれぞれ独立に進化しても、それらの要素を柔軟に取り込む事が可能であり、それによって、各技術の最新のものを利用可能なアーキテクチャとなっていなければならない。

VNode のアーキテクチャは、仮想ネットワークの構造を定義するためのリンク資源を管理するリダイレクタ、及びノード資源を提供し、深遠なプログラム性 (Deeply Programmable) を提供するプログラマの2つの部分に分離し、両者は独立して進化可能となっている。

また端末からエッジ・コアネットワーク全てを包含し全体を仮想化し、スライスの概念をインフラ全体で実現するアーキテクチャとするため、十分なりソースによる資源割当てと高精度の帯域保証が可能なコアネットワークと様々な形態のエンド端末をスケラブルにスライスに収容しフレキシブルに仮想化技術が導入できるエッジ・ネットワークとの2つの要求条件が異なるネットワークで構成した。エッジ・ネットワークの役割は、エンド・ツー・エンドで、新しいプロトコルを使用可能とし、簡単に、仮想ネットワークに無線モバイル端末も含めたユーザ端末収容とクラウドネットワークとの結合を前提条件としてアーキテクチャ検討を行った。

図1は、VNode システムをコアネットワークとしエッジ・ネットワークを結合したスライサブルなエッジ・コア統合ネットワーク仮想化基盤の全体を示している。VNode については、以下に続く各節にて詳しく述べる。エッジ・ネットワーク向けには、VNode と比べて軽量かつ低消費電力のネットワーク仮想化ノードとして FLARE というノードを開発した。FLARE ノードのアーキテクチャの要求条件として、ユーザ端末あるいはアプリケーションの要求に応

じ、ネットワーク処理能力の割当てが行え、最適なネットワーク・プロトコルあるいはネットワーク処理がプログラマブルに提供できることを目指した。

FLARE ノードのハードウェアは、パケット処理を行うデータプレーンにメニコアのネットワークプロセッサを、パケット処理を制御あるいは高度な計算処理を行うコントロール・プレーンには、x86 アーキテクチャのインテル・プロセッサを PCIe インタフェースで結合して構成した。図2に示すような外部の FLARE ノード管理サーバにより、FLARE ノードに、軽量 Linux コンテナ (LXC) 技術による仮想マシンを使用して、両プロセッサ上で仮想化を行い、スライス生成する構成になる。メニコア・プロセッサ上でのスライス生成では、スライスごとにコアを割り当てることで、スライス間の処理のアイソレーションを確保している。また物理ポートに対し、入出力ポートあるいはパケットに付加したタグ情報にて、パケット分離して各スライスに振り分ける Slicer スライスと呼ぶスライスを設け、各スライスで個別のネットワーク処理を可能にした。図2では、FLARE ノード1台で、スライス1にイーサネット・スイッチ、スライス2に MAC アドレス・バイト長を拡張したパケットをスイッチングするスイッチ、さらにスライス n に OpenFlow スイッチの各機能をプログラマブルに設定可能であることを示している。

エッジ端末あるいはエッジ・ネットワークからのトラフィックをコアネットワークのスライスに収容するためアクセスゲートウェイ (AGW) の開発を行った。図3(A)に示す IPSec トンネル及びタグ VLAN でスライスに収容する方法と、図3(B)に示すような、パケットのトレイラーに情報バイトを付加する方式を開発した。トレイラー・バイト方式では、既存ネットワークが含まれていても、FLARE ノードでトレイラー・バイトと VLAN ID 変換処理を行うことで、宛先のコアネットワークのスライスに収容できることが特徴である。さらにトレイラー・バイトに端末で使用して

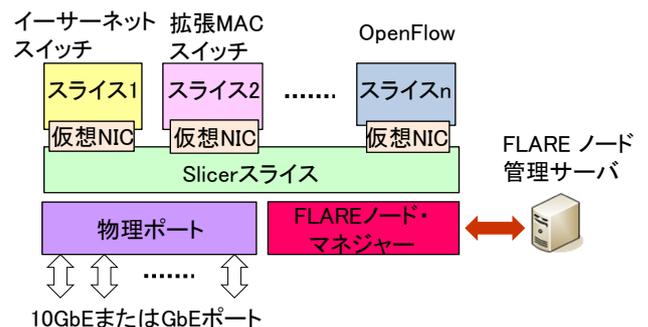


図2 FLARE ノードの構成

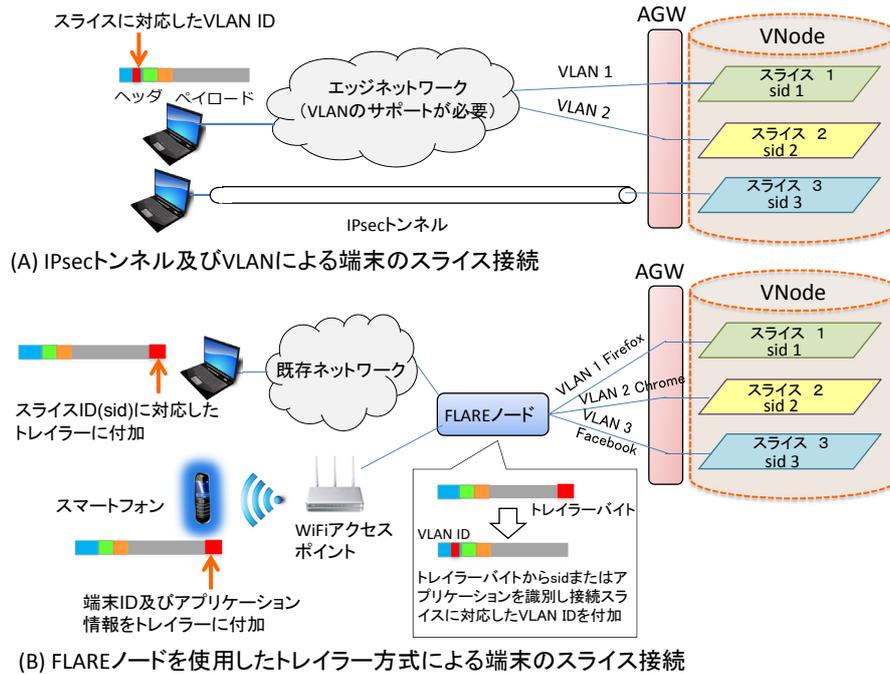


図3 エッジ端末トラフィックのスライス収容方法

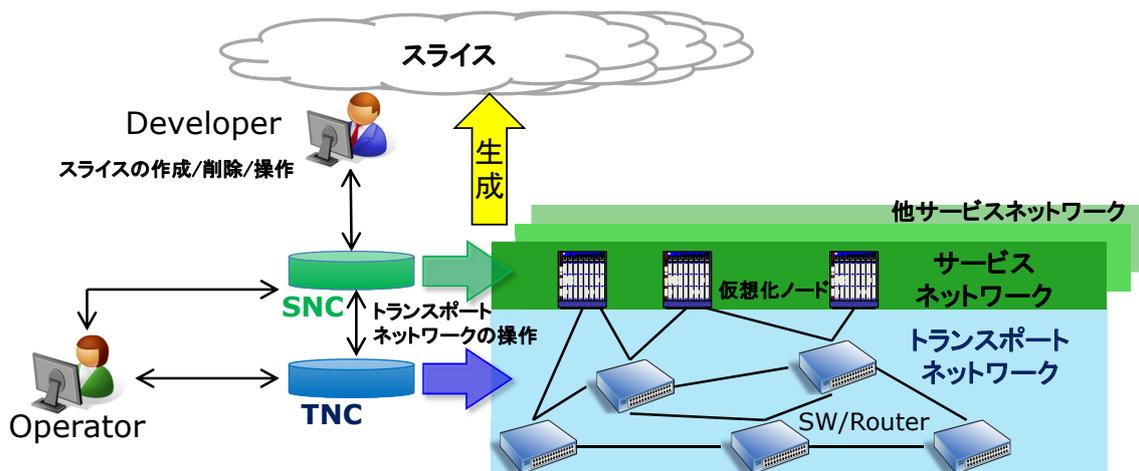


図4 サービスネットワークとトランスポートネットワークの構成

いるアプリケーション情報と端末識別IDを書き込む処理を端末内で行うことにより、従来はDeep Packet Inspection (DPI) 装置が必要となるアプリケーション特化したスライス収容が、容易にできるようになった。

3.2 制御管理機構(資源の抽象化と融通性)

統合管理型ネットワーク仮想化基盤では、多様な物理ネットワーク資源を抽象化し、仮想化することにより、柔軟かつ統合的な資源管理を実現している^[2]。本研究開発では、資源の抽象化管理をトランスポートネットワークへ拡張し、トランスポートネットワークとの連携制御管理を実現した。一般的なネットワークサービスを提供するインフラネットワークの構成

は、ネットワークサービスを提供するサービスノード群から成る“サービスネットワーク”とサービスノード間を接続しそのパケットを転送する“トランスポートネットワーク”から成り、後者はサービス内容によらずパケットを転送し、複数のサービスで共用される。ネットワーク仮想化基盤でもVNodeで構成されるサービスネットワークと、既存のスイッチやルータで構成されるトランスポートネットワークにて構成されるため、ネットワーク仮想化基盤管理システムには、サービスネットワーク制御管理とトランスポートネットワーク制御管理の統合制御管理が必要となる。図4にサービスネットワークとトランスポートネットワークの統合制御管理の構成を示す。サービスネットワー

クの制御管理を行うサービスネットワーク管理システム (Service Network Controller: SNC) と、トランスポートネットワークの制御管理を行うトランスポートネットワーク管理システム (Transport Network Controller: TNC) とを連携させ、スライス生成時に仮想リンクを実現するために必要なトランスポートパスへの資源割当てやトランスポートパスの自動生成を行う。SNC ではトランスポートネットワーク資源を抽象化されたトランスポートパスとして扱い、仮想リンクを作成するために必要なトランスポートパスの資源量を TNC へ通知する。TNC では用意された物理資源を利用して必要なトランスポートパスを生成し、SNC に対して提供する。

資源の融通性として、スライスに割り当てられているネットワーク資源を動的に変更し、サービス品質の維持や全体最適化を実現するためには、スライス自身の性能の測定が必要となる³⁾。ネットワーク仮想化基盤では、スライスごとにスライスのトポロジや識別 ID が異なるため、SNC 及び TNC から測定するスライスのトポロジ情報とスライス情報を取得し、測定点やフィルタ条件の設定を行う NW Monitoring Manager を開発した。図 5 にネットワーク仮想化基盤管理システムと NW Monitoring Manager との連携によるスライス計測方法を示す。スライスの管理者である Developer は、測定したいスライス、測定地点、測定日時をネットワーク仮想化基盤管理システムの Portal 画面から指定する。ネットワーク仮想化基盤管理システムからは、測定リンク情報とスライス識別情報を NW Monitoring Manager に提供し、NW Monitoring Manager にて測定点を切り替える光スイッチと計測を行う PRESTA 10G のフィルタ設定を行う。測定結果は NW Monitoring Manager からネットワーク仮想化基盤管理システムを介して Developer に通知される。

3.3 リソース制御 (資源の独立分離性)

ネットワーク仮想化においてはスライス間のリソース干渉を防ぐことが重要である。たとえば、あるスライスが通信帯域を独占したり他のスライスに不当な遅延を発生させたりしてはならない。このようなリソース干渉を避ける機能が「リソースアイソレーション」である。端点間 (end-to-end) で干渉をなくすには仮想化基盤全体でリソースアイソレーションを達成する必要があるが、この目標を達成する上で特に重要なのはリダイレクタであると我々は考えた。従来のネットワーク仮想化基盤でもリダイレクタにおいて基本的なリソースアイソレーション機能が実現されていた⁴⁾が、今回の研究開発においてそれを更に強化した。従ってここではまず基本機能について記述し、本研究開発における強化点について記述する。

リダイレクタにおけるリソースアイソレーション基本機能については論文[4]に記述したが、その概要は次のとおりである。リダイレクタは VLAN 機能、IP ルーティング機能、QoS 機能などをもつ L3 スイッチを内蔵している。このスイッチは QoS 機能として Weighted Fair Queuing (WFQ) 機能、ポリシング機能などをもっているが、リダイレクタはこれらを使用してリソースアイソレーションを実現している。WFQ 機能はスイッチからパケットを出力する際にパケットをフローごとにキューイングしてフローの出力帯域を制御するが、リダイレクタはこの機構によってスライスごとの高性能な帯域制御を実現している。しかし、この方法はキューイングのために高速メモリが必要なため高速回線においては比較的高コストであり、従来は限られたスライスだけがこの機能を使用できた。これに対してポリシング機能はメモリを必要としないため低コストであり、数十個以上のスライスにおいてリソースアイソレーション機能を利用することができた。ただし WFQ を使用する方法と比べると性能は低

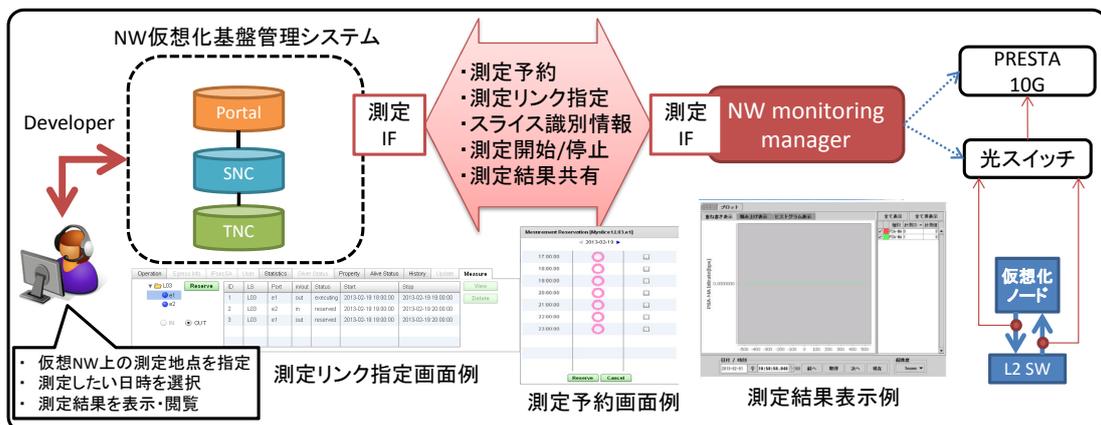


図 5 ネットワーク仮想化基盤管理システムと NW Monitoring Manager の連携

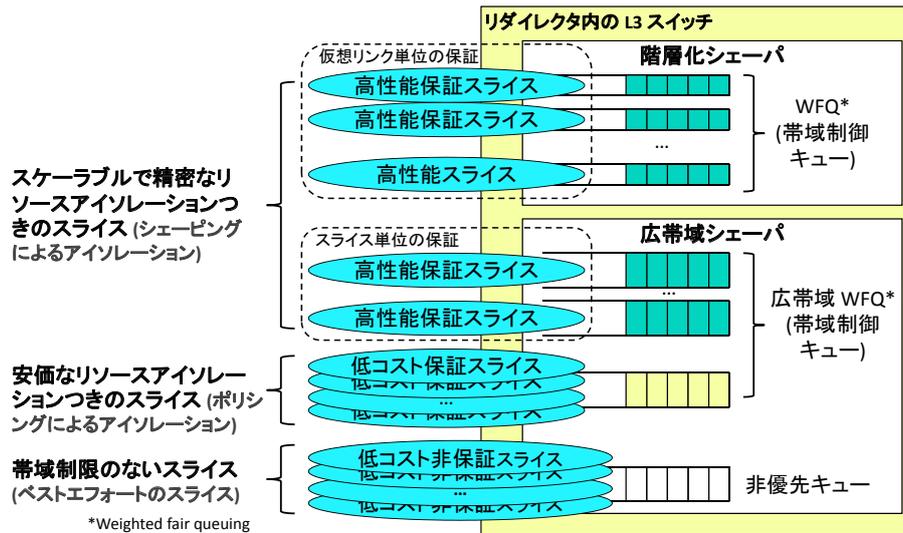


図6 リダイレクタのリソースアイソレーション機能

下する。すなわちパケット損失や他のスライスのジッタ増加が発生しやすくなる。

上記の基本機能の課題は3つあった。第1の課題はWFQに基づくリソースアイソレーションをより多くのスライスに提供するとともに、1個の仮想ノードに接続された仮想リンクについては総計で帯域が制御されていたリソースアイソレーション機能⁴⁾を、仮想リンク単位で制御するように改善してより精密な制御を実現することだった。その実現のためリダイレクタの内蔵スイッチに階層化シェーパとよばれるハードウェアを搭載し、その機能を利用してスケラブルで精密なリソースアイソレーション機能を提供できるようにした(図6)。第2の課題はリダイレクタにおけるリソースアイソレーション機能がVNodeの他の部分(特にプログラマ)や仮想化基盤の他の要素であるAGWやトランスポートネットワークと連携していなかったために困難だった端点間の干渉をなくすことである。解決のためプログラマ-リダイレクタ間でのリソース管理の連携を実現するとともに、ネットワーク仮想化基盤管理システムにてリダイレクタ-AGW間やトランスポートネットワークの資源管理を行うことにより、課題解決を達成した。第3の課題はVNode同士の連携である。この課題解決は3.4.2において記述する「プラグインアーキテクチャ」の使用によって解決への一歩を踏み出した。すなわちスライス上の機能には影響を与えずにネットワーク仮想化基盤におけるパケットの遅延を計測する実験を成功させた⁵⁾。これは、限られた目的ではあるがネットワーク仮想化基盤の全VNodeがリソース制御のために連携できるようにしたことを意味している。

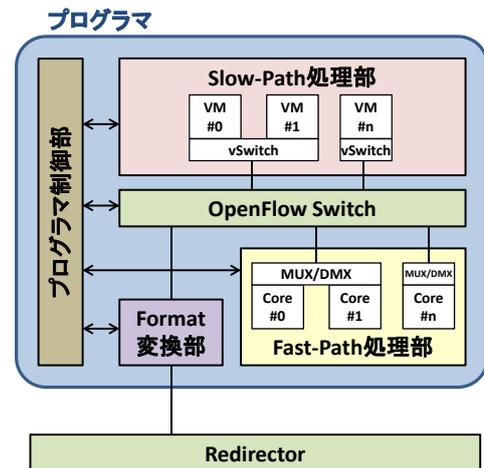


図7 プログラムの構成

3.4 プログラム性拡張(プログラム可能性)

3.4.1 プログラム性とパフォーマンスの両立

VNodeを構成するコンポーネントの1つであるプログラマは、ネットワーク処理におけるプログラマビリティとパフォーマンスの両立を実現することを目的としている。

この実現のために、プログラマでは、汎用サーバ上の仮想マシンによる柔軟なプログラミング環境を提供するスローパスと、ネットワークプロセッサによる高いパケット転送性能を有するプログラミング環境を提供するファストパスの2種類の機構を準備することで、多様なネットワーク機能の実現への対応を行った。さらにスローパスに関しては、高い計算性能に対するネットワークI/O性能のギャップに起因するパフォーマンス上の課題解決を行った⁶⁾。

プログラマの構成を図7に示す。

4 新世代ネットワーク基盤技術

スローパスは、仮想マシン (VM) による柔軟なプログラミング環境と、後述する性能向上の実現による高速なネットワーク I/O 環境を提供する。一方ファストパスは、ネットワークプロセッサによる高いパケット転送性能を有するプログラミング環境を提供する。装置内の計算機資源の接続には OpenFlow Switch^[7] を用いることで、各種計算機資源を自由に組み合わせ可能なネットワークノードを構築することができる。Format 変換部は、リダイレクタとプログラマ間のパケットフォーマット (MAC-in-MAC と VLAN フォーマット) の変換を行う。

スローパスに対しては、図 8 に示すように① vSwitch オフロード技術の適用、②ハードウェアの 10GbE 化により、ネットワーク I/O 性能の向上を実現した。図 9 に、上述の①と②の対応を行ったスローパスの性能評価結果を示す。onload が対応前の、offload が対応後の結果である。CPU Load のSUM はゲスト OS の CPU 使用率の合計で、HOST はホスト OS のみのものである。この図より、スループット性能が向上し、かつホスト OS の CPU 使用率も低下していることが分かる。

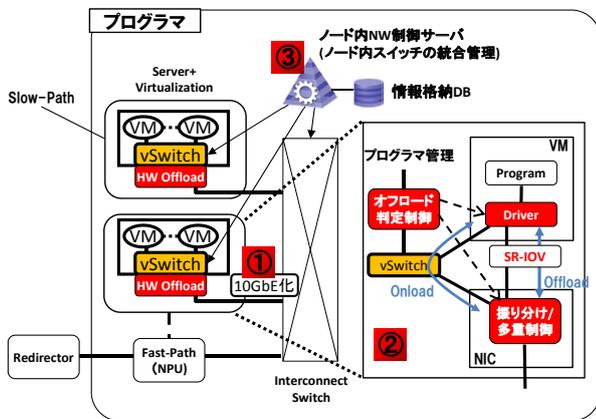


図 8 SlowPath の性能改善

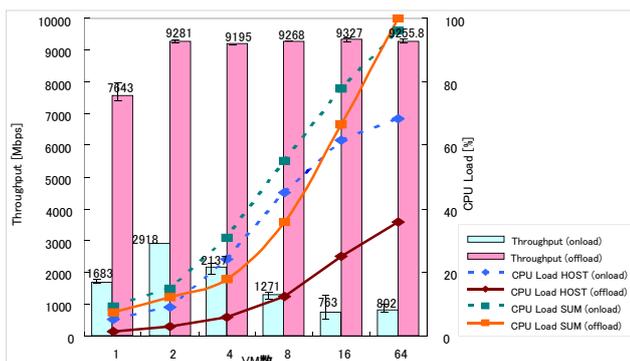


図 9 SlowPath 性能 (受信時) 向上結果

3.4.2 プラグインによるプログラム性拡張

プラグインによるプログラム性拡張の研究開発には2つの目的がある。第1の目的はソフトウェアの追加により VNode を進化させられる (機能拡張できる) ようにすることである^[8]。VNode においてはあらかじめ決められた種類の仮想ノードと仮想リンクとが使用できる。仮想ノードとしては前述したようにスローパスとファストパスという2種類 (2つの型)、仮想リンクとしては Generic Routing Encapsulation over IP (GRE/IP) に基づく1種類 (1つの型) が使用できる。スローパスは自由にプログラムできるが、目的によっては他の種類の仮想ノードや仮想リンクを導入してプログラム性を拡張することが必要になる。このような場合に VNode のインフラストラクチャを拡張して新たな型の仮想ノードや仮想リンクが組み込めるようにすることが目的である。第2の目的はインフラストラクチャの機能拡張にあたってそのプログラミングを容易にするための方法を開発することである。すなわち VNode にネットワークプロセッサ等の特殊化されたハードウェアを追加することによって機能拡張するとき、開発者にとってそのプログラミングの困難さが機能拡張の障害となりうるが、この問題を解決することが目的である。

これらの目的を達成するために次の3項目を試作した (図 10)。第1はオープンなファストパス処理基盤の言語と開発環境の試作である^[9]。この試作は第2の目的にかかわり、従来は高級言語によってプログラムすることが困難だったネットワークプロセッサ等の特殊なハードウェアのために容易に高性能なプログラムを記述することができる言語 Phonepl とその開発環境を部分的に開発した。第2はファストパス処理機能の VNode への組み込み機構の試作である^[8]。この試作は第1の目的にかかわり、第1の試作項目であるファストパス処理基盤を使用して開発したファストパスモジュールや他の種類のファストパスモジュールを

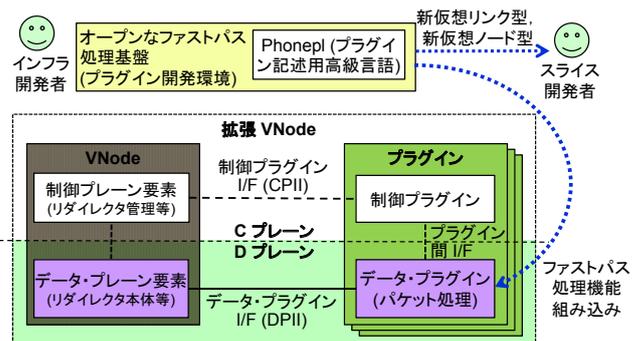


図 10 プラグインによるプログラム性拡張機能の試作内容

VNode に組み込むこと、つまり実行環境の構築を可能にした。第3はスイッチ／ルータ機能オフロード機能のVNodeへの組み込み機構の試作である^[10]。この試作も第1の目的にかかわり、VNode特にその一部であるリダイレクタが持ちながら開発者からすなわちスライス上からは利用できない機能をモジュール化して開発者に提供することを可能にした。

3.5 エッジ仮想化技術 ネットワークアクセスのプログラム技術

エッジ仮想化技術においては、コアネットワークからエッジ・ネットワークまでスライスを広げ、ネットワーク構造における最適な場所でのネットワーク新機能の実装を可能とすることで、より効率の良いネットワークサービスの展開と、アプリケーションに特化したネットワーク機能の提供を目指している。この目的のため、本節では移動通信端末のような特性の異なる複数のネットワークを持つ端末を対象として、特性の異なるアクセスネットワークに対する接続を動的に切り替え、かつ、その切り替えをネットワーク側から制御することで、VNode上で展開されるサービスと連携可能な端末ネットワークアクセス動的制御基盤の研究開発について記す^[11]。

今回実現した端末ネットワークアクセス動的制御基盤を図11に示す。

Edge NodeはEdge SNCからのポリシー指示に基づき、アプリケーションごとに通信可能なスライスを分離し、スライスごとにアクセスネットワークとしていずれの種類の無線ネットワークを利用するかを制御可能である。今回Edge Nodeは、android端末上に実装した。

Edge Gatewayは、エッジ・ネットワークスライスとコアネットワークスライスとの通信を中継するゲ

トウェイ装置である。コアエリア上のゲートウェイ装置であるAGWに接続し、VNode上のNode SliverとEdge Node上のアプリケーションとの通信を中継する。Edge SNCは、エッジ・ネットワークスライスの生成・削除・設定、VNode上のコアネットワークスライスとの結合の実行指示、Edge Nodeの認証を行う。さらにEdge SNCは、VNodeの統合管理装置であるSNCと通信し、エッジ・ネットワークスライスとコアネットワークスライスとの接続に必要な情報をやり取りする。

これらの機構により、Edge Node上のアプリケーション単位で、任意のアクセスネットワークを用いて、エッジ・ネットワークスライスとコアネットワークスライスを結合したスライスへの収容を実現した。

3.6 ゲートウェイ機能強化(認証性)

ネットワーク仮想化基盤上のスライスを利用するには、実ネットワークも含めたend-to-endの接続性が必要である。さらに本仮想化基盤の特長上、独自も含めた様々なプロトコルで接続可能であることが求められる。この接続性を提供する装置がゲートウェイ(AGW: Access Gateway)である(図12)。AGWはスライスのエッジに設置され、以下の特徴を有する装置である。

- Connectivity: 物理端末／ネットワークとスライスを接続する仮想網振分機能
- Security: 物理端末接続時にIPSecを用いた認証機能を提供
- Programmability:
 - GW機能部 Packet processing middleware (PPM)のプログラマブルなプロトコルスタックにより、多種のプロトコルに対応したカスタマイズが可能

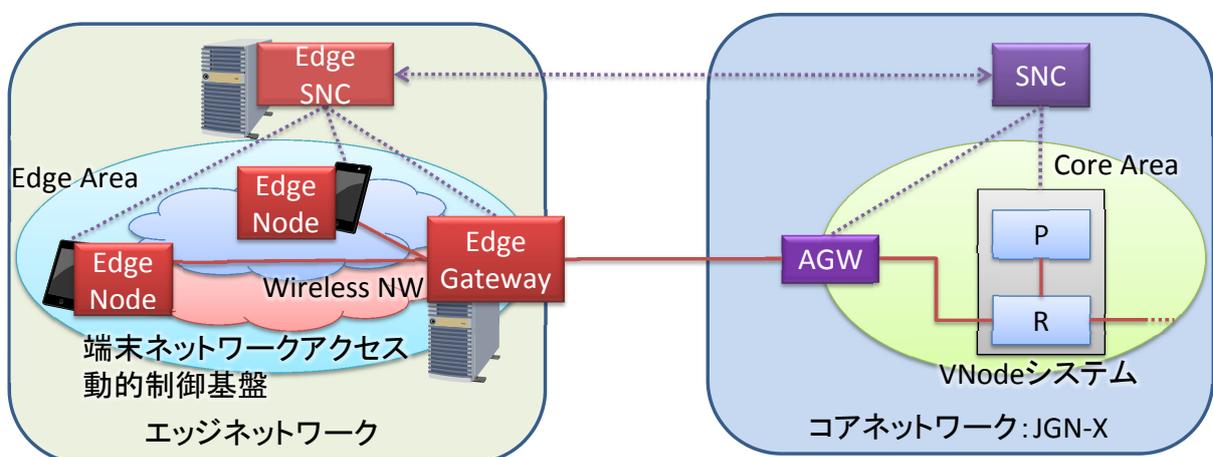


図11 端末ネットワークアクセス制御基盤

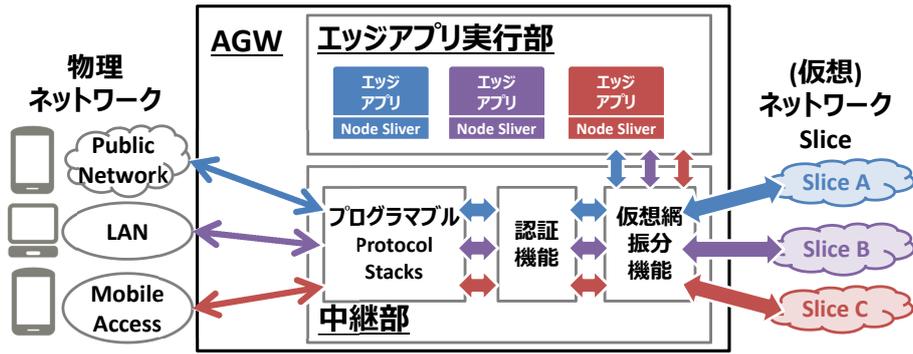


図 12 ゲートウェイの位置づけと提供機能

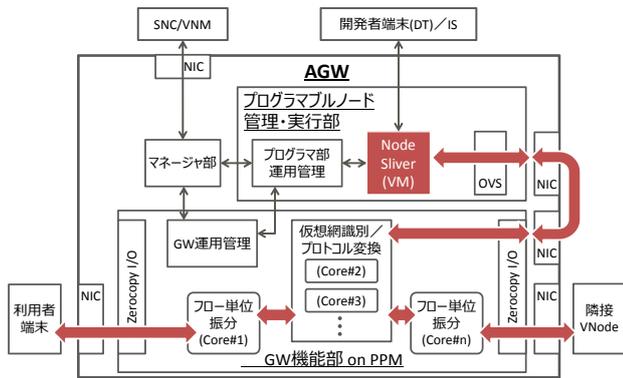


図 13 ゲートウェイアーキテクチャ

➤ プログラマブル仮想ノード (Node Sliver VM) の提供により、スライスの入り口(エッジ)で任意の通信処理アプリが動作可能

- Performance: GW 機能部 PPM により、汎用 IA サーバで高い中継処理性能を提供

図 13 にはゲートウェイ装置の内部アーキテクチャと内部通信形態を示す。ゲートウェイは、スライスのエッジ部分に設置される装置であり、既存 VNode の様な大型の装置ではなく、小型・安価な装置とする必要がある。このため、ゲートウェイの最小基本構成を、IA サーバ 1 筐体で全ての機能(既存ゲートウェイ機能とプログラマビリティ機能)を提供することとしつつ、必要に応じて装置を追加することでプログラマビリティ機能部に関してスケール可能とすることをターゲットとした。これを踏まえたゲートウェイは大きく以下の 3 つのコンポーネントから構成される。

- マネージャ部: SNC/VNM と XML-RPC の制御メッセージ通信を行い、中継部とプログラマ部に対して通信設定・解除、状態問い合わせなどを行う。
- プログラマブルノード管理・実行部: Developer が開発したプログラム (SlowPath-VM) を実行し、転送フレームに対するプログラマビリティ (フレームの判別・変換・加工等) を提供する。

表 1 ゲートウェイ性能

IPSec 転送性能	VLAN 転送性能	SlowPath Node Sliver
1.3 Gbps ^{*1}	4.7 Gbps ^{*1}	1.5 Gbps ^{*2}

*1. 1372 bytes frame, Intel Xeon X5690 (3.46 GHz/6 コア) × 2 個

*2. Intel Xeon X5690 (3.46 GHz/6 コア) を用いて VM に仮想 CPUx2, メモリ 2 GB, vhostnat 処理に 4 コアを割当

- GW 機能部: 利用者端末 (UT) と VNode 間でユーザフレームの送受信を行い、実ネットワークとスライス間のフレーム転送及びフォーマット変換を行う。PPM 機能が提供する Zero Copy I/O、マルチコア CPU 活用並列処理により高速な中継処理を実現。

本ゲートウェイ装置の中継処理性能を表 1 に示す。

4 テストベッド展開と国際相互接続

研究開発を行ったネットワーク仮想化基盤を JGN-X テストベッドへ展開し、アプリケーション実験利用者に提供した。図 14 に示すように、仮想化ノード (VNode) を全国に 7 台、小型仮想化ノード (NC) を 2 台、アクセスゲートウェイ (AGW) を 6 台配備した。1 か月あたり 40 スライス程度の利用実績がある。

また、NICT により米国ユタ大学に設置された小型仮想化ノード (NC) 1 台を介して JGN-X 上に構築したネットワーク仮想化テストベッドと、北米のネットワーク仮想化テストベッド GENI、欧州のネットワーク仮想化テストベッド Fed4 FIRE との相互接続実験を行い、グローバルなマルチドメイン環境でのスライス構築を実証した。図 15 に相互接続実験の構成を示す。各ネットワーク仮想化テストベッドはユタ大学の小型仮想化ノード内に設置した SEP (Slice Exchange

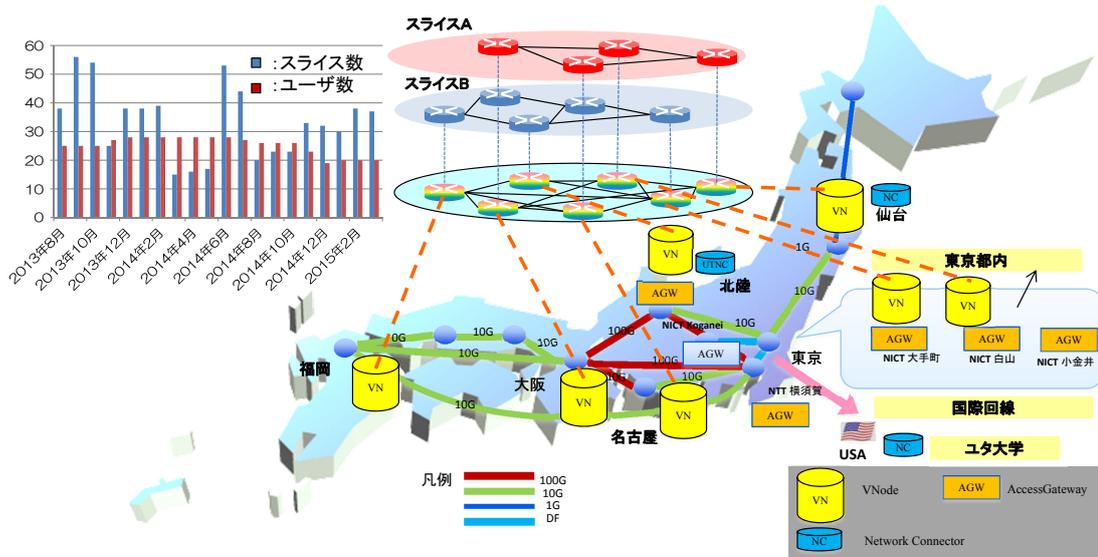
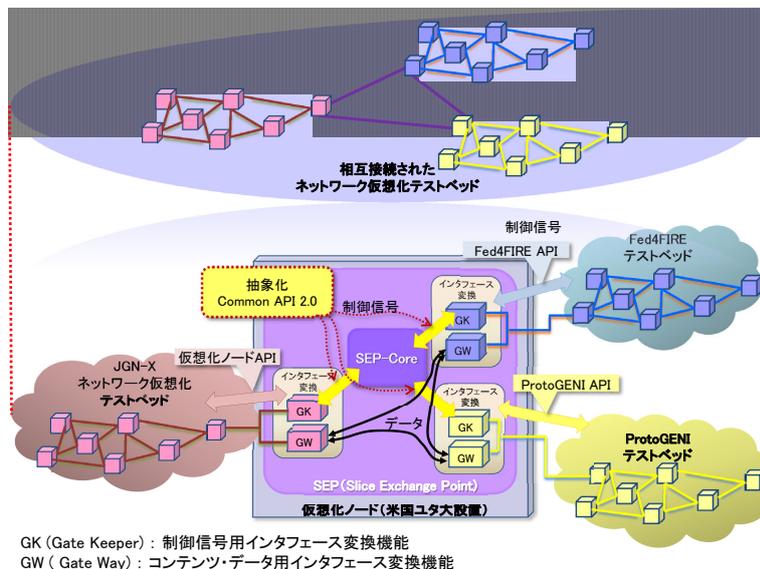


図 14 ネットワーク仮想化基盤の JGN-X 展開



GK (Gate Keeper) : 制御信号用インタフェース変換機能
 GW (Gate Way) : コンテンツ・データ用インタフェース変換機能

図 15 日米欧相互接続構成

Point)を介して接続される。SEPでは、それぞれのネットワーク仮想化テストベッドのインタフェースを共通APIとして抽象化することにより、それぞれの実現方式の差を吸収している。

5 むすび

我々が研究開発を進めてきた統合管理型ネットワーク仮想化基盤は現在の情報通信の多くの課題を解決できる可能性を持っている。今後はテストベッド上での利用実験を通して、新世代の情報通信基盤としての評価を行う予定である。

謝辞

本研究の成果は NICT 委託研究「新世代ネットワークを支えるネットワーク仮想化基盤技術の研究開発」によるものである。

【参考文献】

- 1 中尾 彰宏, "VNode: A Deeply Programmable Network Testbed Through Network Virtualization," 第3回 NV 研究会, Mar. 2, 2012
- 2 Yohei Katayama, Kazuhisa Yamada, Katsuhiko Shimano, and Akihiro Nakao, "Hierarchical Resource Management System on Network Virtualization Platform for Reduction of Virtual Network Embedding Calculation," APNOMS2013, P2-1, 2013
- 3 桑原世輝, 片山陽平, 山田一久 "仮想ネットワークに適應するトラフィックモニタリングシステムの提案," 2013 電子情報通信学会総合大会 B-7-4, 2013
- 4 Kanada, Y., Shiraiishi, K., and Nakao, A., "Network-resource Isolation for

4 新世代ネットワーク基盤技術

- Virtualization Nodes,” IEICE Trans. Commun., Vol.E96-B, No.1, pp.20–30, 2013, DOI: 10.1587/transcom.e96.b.20.
- 5 Kanada, Y., “Extending Network-virtualization Platforms by using a Specialized Packet Header and Node Plug-ins,” 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCom 2014), Sept. 2014, DOI: 10.1109/softcom.2014.7039092.
 - 6 神谷聡史 他, “進化型ネットワーク仮想化基盤を実現するネットワーク処理計算ノードの提案,” 第3回ネットワーク仮想化時限研究会, March, 2, 2012
 - 7 [OpenFlow] OpenFlow Switch Consortium, OpenFlow Switch Specification, version 1.0.0, Dec, 2009.
 - 8 Kanada, Y., “A Method for Evolving Networks by Introducing New Virtual Node/link Types using Node Plug-ins,” 1st IEEE/FIP International Workshop on SDN Management and Orchestration (SDNMO’14), May 2014, DOI: 10.1109/noms.2014.6838417.
 - 9 Kanada, Y., “High-Level Portable Programming Language for Optimized Memory Use of Network Processors,” Communications and Network, Vol.7 No.1, 2015, DOI: 10.4236/cn.2015.71006.
 - 10 Kanada, Y., “Providing Infrastructure Functions for Virtual Networks by Applying Node Plug-in Architecture,” Workshop on Software Defined Networks for a New Generation of Applications and Services (SDN-NGAS 2014), August 2014 & Procedia Computer Science, Vol.34, pp.661–667, 2014, DOI: 10.1016/j.procs.2014.07.094.
 - 11 飯星貴裕 他, “仮想化ノードにおける容易なネットワークサービス実験のためのスマートデバイスへのスライス延伸,” 第9回ネットワーク仮想化時限研究会, April 11, 2014



雨宮宏一郎 (あめみや こういちろう)

株式会社富士通研究所ソフトウェア研究所
分散システムソフトウェアPJ
クラウド、ネットワーク



山田一久 (やまだ かずひさ)

NTT 未来ねっと研究所メディアイノベーション
研究部将来ネットワーキング技術研究グループ
主任研究員
ネットワーク



中尾彰宏 (なかお あきひろ)

東京大学大学院情報学環教授
Ph.D.
ネットワーク、コンピュータサイエンス



金田 泰 (かなだ やすし)

日立製作所情報通信イノベーションセンタ
ネットワーク研究部主任研究員
Ph.D
ネットワーク、コンピュータビジョン



才田好則 (さいだ よしのり)

NEC SI・サービス技術本部技術戦略部エキス
パート
ネットワーク