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

飯星　貴裕[†]　才田　好則[†]　渡邊　義和[†]　森田　弦[†]　狩野　秀一[†]

† NEC 情報・ナレッジ研究所　〒 211–8666 神奈川県川崎市中原区下沼部 1753
E-mail: †{t-iihoshi@cj, y-saida@cp, y-watanabe@fa, g-morita@ax, karino@da}.jp.nec.com

あらまし　サービスの構成要素としてネットワークが重要になってきている．仮想化ノードは，このようなネットワークと強く結びついたサービスアーキテクチャの開発・実証に適している．一方，サービスの利用主体にスマートデバイスが加わったことで，スマートデバイスの特徴を生かしたサービスの開発も重要になっている．そこで我々は，特にスマートデバイスの可搬性を生かしてサービスの開発・実証を行うことのできる仮想化ノードエッジシステムを開発した．本稿では，サービスの開発・実証のケーススタディから設計と実装を議論する．

キーワード　仮想化ノード，テストベッド，スマートデバイス

A Slice Extension Architecture Using Smart-devices for Flexible Network Service Experiments on VNode

Takahiro IIHOSHI[†], Yoshinori SAIDA[†], Yoshikazu WATANABE[†], Gen MORITA[†], and Shuichi KARINO[†]

† Knowledge Discovery Reserch Labs., NEC Corporation　1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211–8666 Japan
E-mail: †{t-iihoshi@cj, y-saida@cp, y-watanabe@fa, g-morita@ax, karino@da}.jp.nec.com

Abstract　Network has been playing an important role in service compornents. VNode is a suitable testbed to develop and demonstrate service architectures which are tightly bound with network. On the other hand, it becomes important to develop services suitable for characteristics of smart-devices, with the increasing utilization of them. Therefore, we made a VNode Edge-System to develop services which is focused on portability of smart-devices. In this paper, we discuss a design and implementation of our system from a case-study of service developing.

Key words　VNode, Testbed, Smart-Device

1. はじめに

サービスアーキテクチャが変遷するにつれ，ネットワーク（以下，NW）がサービスの構成要素としてますます重要になってきている．そのため，新たなサービスアーキテクチャの開発・実証には，コンピューティング環境だけでなく，NW 環境も自由に構成できる環境が必要になっている．

NW 融合型サービスのためのテストベッドに，仮想化ノード [1] [2](注1)がある．仮想化ノードは，自由に NW とサービスコンポーネントを設定できる隔離環境を提供できるため，NW と融合したサービス（以下，NW 融合型サービス）の開発や実証に適している．

一方，これらのサービスを利用するクライアントとして，スマートデバイスが主要なものになってきている．よって，サービスの開発においては，スマートデバイスとの連携も重要になっている．

そこで我々は，仮想化ノードを用いた NW 融合型サービス実験において，スマートデバイスを容易に利用可能にするための仮想化ノードエッジシステムを開発した．

本稿では，スマートデバイスを用いた NW 融合型サービス実験のケーススタディ，ケーススタディから導かれるシステム要件，及びエッジシステムの設計・実装について述べる．

2. 背　　景

2.1 NW とサービスの融合

昨今のサービスの構成要素として，NW はより重要なものと

(注1)：Virtualization Node を略し，VNode とも呼ばれる．

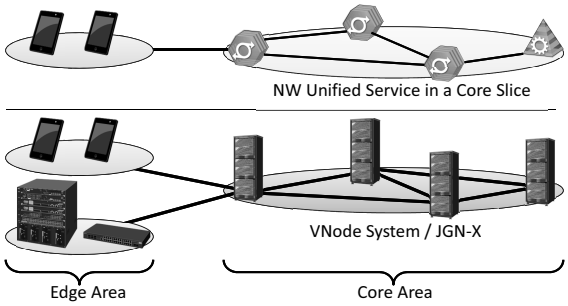

図 1　仮想化ノードの外部システム接続アーキテクチャ

なってきている．例えば，Contents Delivery Network（CDN）のような分散キャッシュサーバによって提供されるサービスでは，クライアント—キャッシュサーバ間の NW 距離によって，リクエストの配送先サーバが決定されている．また，新たなサービス構成手法として，サービスチェイニング（Service Chaining）も注目されている．サービスチェイニングは，異なるノードにサービス要素を分散配置し，それぞれのサービス要素を経由するようにパケットをルーティングしていくことによって，サービス全体を構成するものである．

このような NW と密に結合したサービスの開発・実証を行うとき，環境の確保がしばしば課題となる．サービスの性能特性は，NW のトポロジや帯域に大きく左右される．そのため，それらを自由に構成できる環境が必要となるが，そのような環境を自ら構築するには大きなコストと時間がかかる．

2.2　仮想化ノード

NW 融合型サービスの実験・実証を行うことができるテストベッドに仮想化ノードがある．仮想化ノードでは，NW 仮想化技術により，NW 資源と計算資源をスライスとして提供する．

仮想化ノードのスライスはお互いに影響しないよう隔離されており，スライス上では NW のトポロジと帯域を高い自由度で設定できる上，ルータやスイッチを含めた NW コンポーネントのすべてを自由に開発できる．そのため，NW コンポーネントに新しい機能を追加することの多い NW 融合型サービスの開発・実証に適している．

また，仮想化ノードでは，ゲートウェイ装置を通すことで，スライス上の NW 融合型サービスと手元の機器や NW を接続することもできる．こうして接続される機器や NW は，エッジエリアと呼ばれる仮想化ノード外部に位置付けられている．また，仮想化ノードはコアエリアと呼ばれ，仮想化ノード上のスライスはコアスライスと呼ばれる（図 1）．

2.3　NW 融合型サービスとスマートデバイス

近年，スマートフォンやタブレット端末などのスマートデバイスが爆発的に普及した．これにより，NW 上に展開されるサービスの利用主体として，パーソナルコンピュータ等に加えてスマートデバイスも主要なものになってきている．

スマートデバイスは，パーソナルコンピュータ等にはない特徴がある．まず，スマートデバイスの多くは常に持ち歩くことが前提である．また，多くのセンサを搭載しており，それらをサービスへの入力に用いることは，すでに一般的となっている．

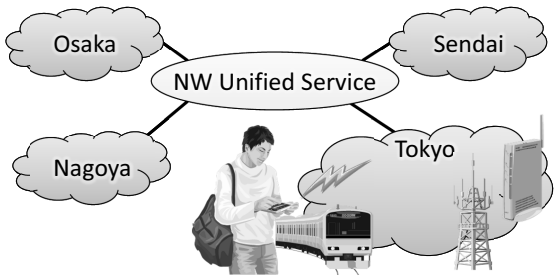

図 2　NW 融合型サービス実験のイメージ

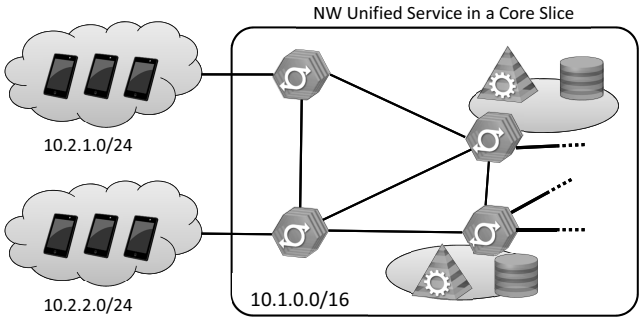

図 3　NW 融合型サービスの実験環境構成の例

そこで本稿では，仮想化ノードを用いた NW 融合型サービス実験において，スマートデバイスの可搬性を生かした実験を容易に実現するためのエッジシステムを検討する．

3.　ケーススタディ

本節では，スマートデバイスを用いた NW 融合型サービス実験についてケーススタディを行う．

スマートデバイスを用いた実験は様々な場所や環境で行うことが想定される．スマートデバイスは多くの場合持ち運ぶことを前提に作られており，携帯電話網や無線 LAN などの無線アクセス NW に接続して通信する．また，スマートデバイスには，GPS やカメラやマイクなどといった様々なセンサが搭載されている．スマートデバイスを用いる実験では，これらの特徴を活用することが想定される．

例えば，センサー情報と連動した NW 融合型サービスを移動しながら利用したり，移動などで不規則に変動する無線アクセス NW の帯域に NW 融合型サービスを最適化したりといった実験内容が考えられる．図 2 に，具体的な実験例のイメージ図を示す．図 2 の NW 融合型サービスの例では，広域 NW 上のコンテンツ配信サービスを想定している．

図 3 に，実験環境の構成例を示す．図 3 の構成例では，スマートデバイスが接続されている NW をそれぞれ異なる NW ノードに接続している．例えば CDN のように NW 距離が重要な NW 融合型サービスの実験においては，異なるルータの先にクライアントを配備することがある．また，図 3 の構成例では，環境全体のアドレス空間を，エリアによってサブネットに分けている．NW 融合型サービスの実験においては，ルーティングが非常に重要な要素になることがある．これを自由に構成するため，実験環境全体のアドレスを自ら決定する．

4. 要件分析

本節では，3. 節で述べたケーススタディと仮想化ノードの設計をもとに，本稿におけるエッジシステムの要件を検討する．

4.1 条件整理

本稿におけるシステムの要件分析を行うにあたり，前提条件について整理する．

本稿で検討するシステムは，前述したように，仮想化ノードのコアスライス上に展開された NW 融合型サービスをスマートデバイスから利用する実験に用いる．そのため，本稿のシステムは仮想化ノードに外部から接続することになり，仮想化ノードのエッジエリアとして位置づけられる．

また，NW 融合型サービスの実験においては，前述したように，仮想化ノードが提供するコアスライスの特徴が重要となる．コアスライスは他のスライスの影響を受けないよう隔離されており，スライス内の NW トポロジやアドレッシング，NW コンポーネントの機能を自由に構成できる．また，コアスライスは複数作成でき，それぞれでサービスを構築できる．

さらに，スマートデバイスを用いた実験では，前述したように，移動を考慮する必要がある．特に，可搬性が高いことから，移動範囲は屋外や路上等にも及びうる．

4.2 要　　件

前節までの整理をもとに，本稿では 7 つの要件を抽出した．以下にそれぞれ説明する．なお，前節の整理と要件の対応関係は，下記に示す通りである．

- 仮想化ノードへの接続: 2
- 実験環境の設計自由度: 1, 3, 4
- スマートデバイスの可搬性への対応: 5, 6, 7

第 1 に，**コアスライス間の隔離を阻害してはならない．**コアスライス間では，パケットが混ざったりアドレスが衝突したりしないようお互いに隔離されている．そのため，それぞれのスライス上で自由にサービスを構築できる．エッジシステムが接続することにより，それらの隔離が阻害されるようなことがあってはならない．

第 2 に，**仮想化ノードの接続インタフェースに準拠しなくてはならない．**仮想化ノードには，外部からコアスライスへ接続するための外部接続インタフェース（認証・データプロトコル）が定義されている．そのため，スマートデバイスからコアスライス上のサービスへアクセスするためには，コアエリアの外部接続インタフェースに従って接続する必要がある．

第 3 に，**複数のコアスライスと接続できるべきである．**仮想化ノードを用いると，性質の異なる NW 融合型サービスを複数のコアスライスでそれぞれ構築できる．そのため，これらのサービスをそれぞれ独立に実験したり，同時に実験したりすることも考えられる．よって，スマートデバイスでは，それぞれ異なるコアスライス上のサービスを同時に利用できることが望まれる．ただし，前述したように，複数のコアスライスと接続しても，各コアスライス上のサービスに影響してはならない．

第 4 に，**エッジエリア内のアドレスを自由に決定できなくてはならない．**3. 節で述べたように，NW 融合型サービスの実験においては，エッジにおいてもノードのアドレスを決められる必要がある．

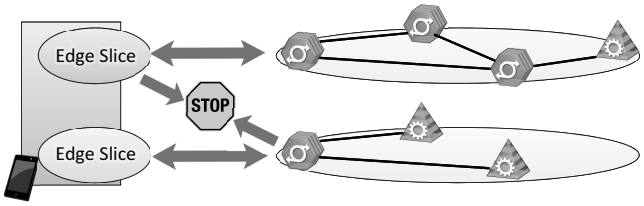

図 4　エッジスライスとコアスライスの接続によるスライス延伸

第 5 に，**スマートデバイスは公衆回線を利用できなくてはならない．**3. 節でも述べたように，スマートデバイスを用いた実験では屋外などで実施されることも想定される．そのような場合，携帯電話網や公衆無線 LAN などのインフラを活用することが想定されるが，環境によっては NAT がかかる場合もある．よって，スマートデバイスによる通信は NAT を越えられる必要がある．また，公衆回線ではセキュリティレベルが不十分な場合もあるため，暗号化も必要である．

第 6 に，**スマートデバイスのエッジエリアへの参加・離脱は動的に行えなくてはならない．**スマートデバイスを用いる場合，いつでもエッジシステムに接続しているとは限らない．例えば，実験を行う時間以外はデバイスの電源が投入されないことが考えられるし，実験内容によって使用するデバイスの数が異なることも考えられる．そのため，スマートデバイスは，エッジシステムに対して動的に参加・離脱を行えなくてはならない．

第 7 に，**無線アクセスネットワーク接続状態の変化に対して，対応できるべきである．**スマートデバイスは，携帯電話網や無線 LAN などの複数の無線アクセス NW に接続でき，移動などによってごく短時間切断されたり，接続している NW が変わったりすることがある．このような接続状態の変化があっても，コアエリアとの接続は極力維持されるべきである．

5. 設　　計

4. 節で述べた要件をもとに，本稿におけるエッジシステムの設計について論じる．

5.1 仮想化ノードとの接続形態

本節では，4. 節で述べた要件から，エッジシステムと仮想化ノード上のコアスライスとの接続形態について議論する．

4. 節で述べたように，本稿のシステムはエッジエリアとして仮想化ノードのシステムに接続する．エッジシステムとしては，複数のコアスライス上のサービスを同時に利用できることが望まれる（第 3 の要件）．一方で，コアスライス間の隔離は保たれなければならない（第 1 の要件）．

これらの要件から，本稿のシステムでは，エッジエリアにおいてもスライスを構築し，これらをそれぞれコアスライスと接続することにより，各コアスライスとの通信を分離する（図 4）．これは，コアスライスをスマートデバイスまで延伸するとも言い換えることができる．なお，エッジエリアにおけるスライスを，本稿ではエッジスライスと呼ぶ．

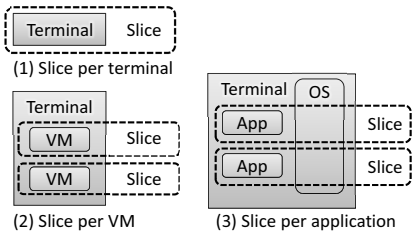

図 5　文献 [2] のエッジスライス提供モデル

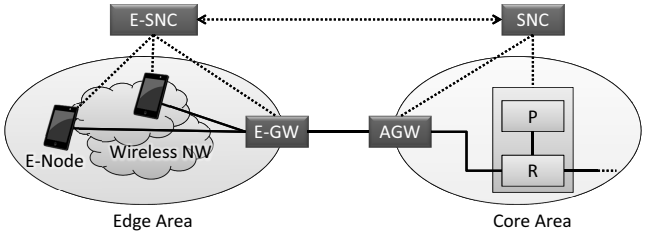

図 6　エッジシステムのアーキテクチャ

5.2　エッジシステムの機能設計

本節では，エッジエリア内で提供すべき機能について検討する．

5.2.1　コアスライス—エッジスライス接続機能

エッジシステムにおけるコアスライスとの接続機能について検討する．

前述したように，本稿のエッジシステムと仮想化ノードとの間には性質のギャップがある．本稿のエッジシステムでは装置の動的な参加・離脱があるなど，仮想化ノードとは異なった性質を持つ（第 6 の要件）．一方，仮想化ノードへ接続するためには，決められた接続インタフェースを使用しなくてはならない（第 2 の要件）．

そこで本稿のシステムでは，エッジシステムと仮想化ノードとの性質差異を吸収しつつ，仮想化ノードへ接続する機能を搭載する．

5.2.2　スライス提供機能

エッジシステムのスライス提供モデルについて検討する．

エッジシステムにおけるスライス提供モデルは，文献 [2] において図 5 のように定義されている．

本稿では，3 つ目のモデルを採用した．1 つ目のスライス提供モデルでは，装置全体を 1 つのスライスに紐付ける．複数のコアスライス上のサービスを同時に利用できないため，本稿のエッジエリアには適さない．また，2 つ目のスライス提供モデルには，比較的大きなリソースを必要とする．スマートデバイスでは動画再生などの高負荷な処理も想定されることから，可能な限り軽量なスライス提供モデルが本稿のエッジシステムでは望ましい．そこで本稿では，アプリケーションごとにスライスへ紐付けるモデルを採用することとした．

5.2.3　エッジスライス内設定機能

エッジシステムのスライス内設定機能について検討する．

本稿におけるエッジシステムでは，各スマートデバイスのエッジスライスにおけるアドレスを設定する機能が必要である．これは，第 4 の要件として述べたように，NW 融合型サービスの実験においてはアドレッシングに自由度が必要なためである．

なお，スライス内設定としてはコンポーネントの配置もあるが，本稿においてはアプリケーションのインストールに相当し，従来のスマートデバイスが持つ機能で十分であると考えられる．

5.3　エッジシステムのアーキテクチャ

本節では，本稿におけるエッジシステムのアーキテクチャについて論じるとともに，その構成要素の概要について述べる．

図 6 に，本稿におけるエッジシステムの構成図を示す．前節で述べたように，本稿のシステムではエッジスライスを構築する．そこで本稿では，仮想化ノードのアーキテクチャを参考にエッジシステムを構成することとした．

本稿におけるエッジシステムは，Edge Service Network Controller（以下，E-SNC），Edge Gateway（以下，E-GW），Edge Node（以下，E-Node）で構成される．

5.3.1　E-SNC

E-SNC は，エッジシステムの統合管理を行う装置である．E-SNC は，E-GW 及び E-Node を制御し，エッジスライスの生成・削除・設定や，VNode 上のスライスとの接続の実行を各装置に指示する．また，E-SNC は E-Node の認証を行い，エッジシステムへの参加を管理する．さらに，E-SNC は，仮想化ノードの統合管理装置である Service Network Controller（以下，SNC）と通信し，エッジスライスとコアスライスとの接続に必要な情報をやり取りする．

5.3.2　E-GW

E-GW は，エッジスライスとコアスライスとの通信を中継するゲートウェイ装置であり，エッジエリアとコアエリアの差異を吸収する機能を持つ．コアエリアにおける外部とのゲートウェイ装置である Access Gateway（以下，AGW）に接続し，コアスライス上の Node Sliver(注2) と E-Node 上の実験用アプリケーションケーションとの通信を中継する。また，E-GW は E-Node の認証を行い，E-GW を介したコアスライスへの接続を管理する．

なお，仮想化ノードにおいて外部からの接続を提供する手段はいくつか提供されているが，複数のコアスライスへ同時に接続するためには，AGW が最も適している．

5.3.3　E-Node

E-Node は，エッジシステムにおいて自身をスライシングする機能を持つノードである．本稿における E-Node はスマートデバイスに相当し，コアスライス上のサービスと通信する．

5.2.2 節で述べたように，本稿の E-Node ではアプリケーションごとに通信可能なスライスを分離する．

5.4　コアエリア—エッジエリア間インタフェース

本節では，コアエリア—エッジエリア間接続の設計について概要を説明する．

SNC—E-SNC 間インタフェースで提供される機能は，(1) **仮想化ノードに対するエッジシステムの認証・認可**と，(2) **エッジシステムから接続可能なコアスライスのリスト，及び各コア**

（注2）：コアスライスにおける計算資源．ルータ・スイッチやサーバ等の各種サービスコンポーネントを実装できる．

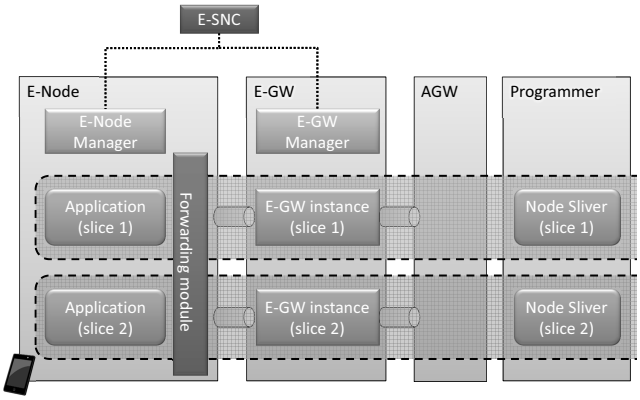

図 7　エッジシステムの実装概要

スライスに接続するための **AGW** 認証情報の通知である．

AGW—E-GW 間インタフェースで提供される機能は，**コアスライス—エッジスライス間 Data Plane の接続**である．このインタフェースは AGW の外部接続インタフェースの仕様に準拠しており，コアスライスごとにセキュアなトンネルを張ることで実現される．各トンネルの認証には，SNC—E-SNC 間インタフェースで通知された AGW 認証情報を使用する．

5.5　Control Plane

本節では，本稿におけるエッジシステムの管理インタフェースについて，その設計概要を説明する．

E-SNC—E-GW 間インタフェースでは，(1)**AGW—E-GW 間インタフェースの接続・切断の制御命令及び AGW 認証情報の配布**と，(2)**E-Node 接続認証情報の配布**が行われる．

E-SNC—E-Node 間インタフェースでは，(1)**E-GW 接続認証情報の配布**と，(2) **アプリケーションとエッジスライスの紐付け情報 (エッジスライス定義) の配布**と，(3) **スライス内における E-Node のアドレス情報の配布**が行われる．なお，E-SNC—E-Node 間インタフェースでは，E-Node—E-GW 間インタフェースの接続・切断を制御しない．これは，本稿におけるエッジシステムでは E-Node が動的に参加・離脱を行える必要があるためである（第 6 の要件）．本稿では，実験者が E-Node 上で直接制御できるようにした．

5.6　Data Plane

本節では，本稿におけるエッジエリアのパケット転送方式について，その設計概要を説明する．

4. 節で述べたように，E-Node—E-GW 間の接続は，暗号化されるとともに，NAT を越えられる必要がある（第 5 の要件）．また，5.3.2 節で述べたように，E-GW において E-Node を認証し，コアスライスへの接続認可を行う必要がある．これを満たすため，本稿では E-Node—E-GW 間には認証と暗号化をサポートするトンネルプロトコルを用いる．

6.　実　　装

本節では，本稿におけるエッジシステム実装の概要を説明する．図 7 に，本稿におけるエッジシステムの実装概要を示す．

6.1　Control Plane

図 7 に示すように，本稿における Control Plane の実装では，E-GW に “E-GW Manager” を，E-Node に “E-Node Manager” を配置する．各 Manager は，5.5 節で述べた情報を E-SNC とやり取りし，各装置内のコンポーネントを制御する．

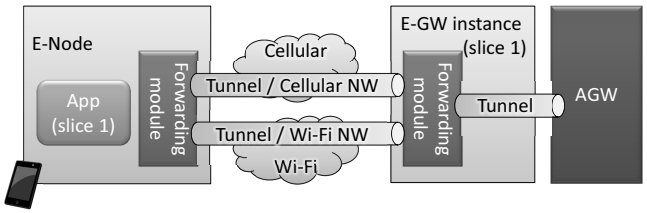

図 8　E-Node – E-GW 間接続

6.2　Data Plane

本稿における Data Plane の実装について説明する．

6.2.1　E-GW

本稿における E-GW の実装では，図 7 に示すように，各スライスに対応する “E-GW instance” を作成する．

E-GW instance では，E-Node—E-GW 間トンネルと E-GW—AGW 間トンネルをそれぞれ終端し，トンネル間でパケットを転送する．E-Node—E-GW 間の接続については，6.2.2 節で後述する．

6.2.2　E-Node—E-GW 間接続

5.6 節で，E-Node—E-GW 間の接続にはトンネルを用いると述べたが，無線アクセス NW 越しのトンネル接続には課題がある．トンネルの確立手続きにかかる時間は NW の遅延に大きく左右される．そのため，無線アクセス NW の切断などに伴うトンネル再確立には 10 数秒かかることもあり，その間はコアスライスとの通信ができなくなる．長時間の通信切断は実験に支障をきたす可能性が高く，4. 節で挙げた第 7 の要件を満たしているとは言い難い．

そこで本稿では，複数の無線アクセス NW 上にあらかじめトンネルを確立しておき，無線アクセス NW の状況に応じて，使用するトンネルを切り替えることとした．E-Node—E-GW 間接続の概要を図 8 に示す．トンネルの切り替えは，E-Node と E-GW において経路選択機能を配備することで実現する．これにより，移動によって無線 LAN が切断されるなどによりトンネルが切断されても，他の無線アクセス NW 上のトンネルで通信を継続できる．本稿における実装では，約 2 秒で切り替え可能となった．

6.2.3　アプリケーション通信の転送

本節では，アプリケーションとスライスの紐付けと，アプリケーション通信のスライスへの転送について，その実装概要を説明する．

まず，アプリケーションとスライスの紐付けを行う方式について説明する．アプリケーションごとに通信するスライスを分けるためには，E-Node において，パケットの送信元アプリケーションを識別し，対応する E-GW instance へ転送する必要がある．そこで本稿では，パケットの送信元ポート番号とオペレーティングシステムの管理情報を用いてアプリケーションを特定し，パケットを各トンネルに振り分けることで実現した．

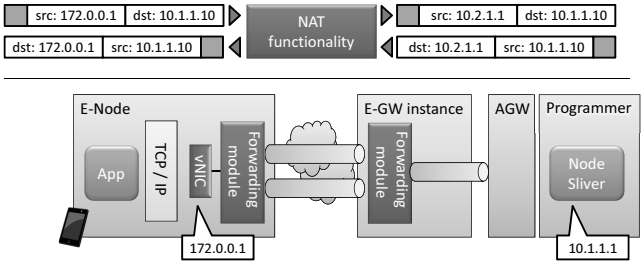

図 9　NAT によるアドレッシング自由度の補完

実装には，文献 [3] と同様に OpenFlow [4] を活用している．

次に，アドレッシングの自由度を保つ方式について説明する．5.2.2 節で述べたように，本稿における E-Node では，アプリケーションごとにスライスを分ける．すなわち，スライスごとに個別のプロトコルスタックを実装せず，単一のプロトコルスタックを複数のスライスで共用する．そのため，コアスライスへ送信するパケットの Src IP アドレスをプロトコルスタック上で自由に設定するのは難しい．そこで本稿では，SNAT 機能を実装し，スライス内設定として変換先アドレスを指定可能にすることにより，スライス内アドレッシングの自由度を補完することとした（図 9）．ただし本方式では，特定のポートで待ち受けるアプリケーションが E-Node 上に複数存在する場合には対応できない．

6.3　エッジスライスプログラミング

本節では，本稿におけるエッジスライスの作成方法と，エッジスライス内のアドレス設定方法について説明する．

6.3.1　エッジスライス定義

本稿のエッジシステムでは，XML の定義ファイルを E-SNC に投入することで，エッジスライスを構築する．定義には，エッジスライスの構築に用いる E-Node の識別情報と，エッジスライスへアクセスする E-Node のアプリケーション情報，接続先 AGW の情報を記述する．

6.3.2　エッジスライス内設定

本稿のエッジシステムでは，XML の設定ファイルを E-SNC に投入することで，エッジスライス内の設定を行う．設定には，コアスライスから見た E-Node のアドレスをそれぞれ記述する．

7.　エッジを含めたスライスの構築例

本節では，コアスライスとエッジスライスで構築できる実験環境について説明する．

図 10 のスライス構築例では，2 つのスライス上にそれぞれサービスが展開されている．スマートデバイス上には，各サービスに対応するアプリケーションがそれぞれインストールされている．

本稿のエッジシステムを用いると，スマートデバイスを含めたサービスの NW をスライスごとに自由に設計できる．本稿のシステムでは，スライス上におけるスマートデバイスのアドレスを実験者が設定できるようになっている．また，スライスとアプリケーションを紐付け，他のスライスと通信を隔離したことにより，意図しないパケットがスライスに入り込むことなどによるトラブルを防ぐことができる．そのため実験者は，他のスライス内の設定を考慮することなく，自身のサービスの実験を行うことができる．なお，図 10 の例では 2 つのスライスで異なるアドレスを用いているが，本稿のシステムは，異なるスライスでアドレス空間が重複する場合にも対応できる．

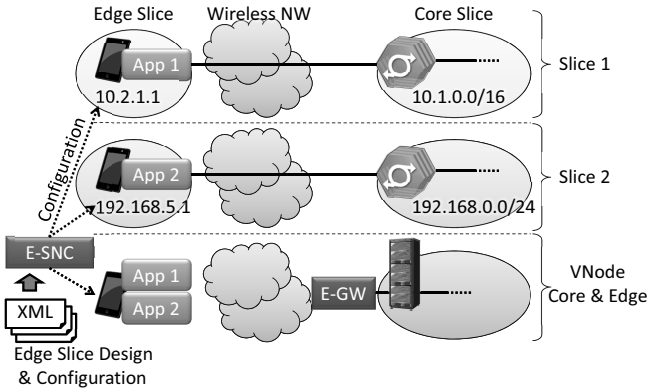

図 10　エッジを含めたスライスの構成例

8.　おわりに

本稿では，スマートデバイスを用いた NW 融合型サービス実験をより容易に実現することを目的とし，仮想化ノードのエッジシステムについて検討を行った．

今後は，仮想化ノードと本稿のエッジシステムを用いて NW 融合型サービス実験を行い，実験の利便性をより改善する予定である．また，仮想化ノードシステムとのよりスムーズな連携に向けた機能検討も行う予定である．

謝辞　本研究は，独立行政法人情報通信研究機構（NICT）の委託研究「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」プロジェクトの成果である．

文　献

[1] A. Nakao, “VNode: A Deeply Programmable Network. Testbed Through Network Virtualization.,” 2012. http://www.ieice.org/ nv/05-nv20120302-nakao.pdf

[2] K. Yamada, Y. Katayama, A. Nakao, S. Yamamoto, Y. Kanada, Y. Saida, and K. Aamemiya, “White Paper VNode: A Deeply Programmable Network Testbed Through Network Virtualization”. https://nvlab.nakao-lab.org/vnode-white.paper.pdf

[3] 飯星貴裕，渡邊義和，森田　弦，才田好則，狩野秀一，“OpenFlow による移動端末通信制御のスケーラビリティ向上方式とその実装 (NS),” 電子情報通信学会技術研究報告 : 信学技報，vol.112，no.463，pp.477–482，mar 2013.

[4] Open Networking Foundation, “Openflow switch specification 1.0.0”.

This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.