Vol.2019-IOT-44 No.49
2019/3/8

国際的なオンデマンドネットワークのための トークンベース認証機構の研究

竹房 あつ子[1,a)]　市川 昊平[2]　栗本 崇[1]　合田 憲人[1]

概要：SINET5 をはじめとする高性能かつ広帯域の学術ネットワークが世界的に配備されており，標準インタフェースから複数学術ネットワークをまたがる VLAN をオンデマンドに構築可能な国際的オンデマンドネットワークが構築されている．しかしながら，認証連携の仕組みとその上でアプリケーションプログラムから制御する方法がないため，計算機等の他の資源との連携が容易ではなかった．本研究では，国際的な研究プラットフォームの構築を加速させることを目指し，アプリケーションプログラムからの認証を可能にする，トークン認証ベースの REST API を開発する．これにより，アプリケーションプログラムから国際的なオンデマンドネットワークの制御が可能になり，他の資源との連携も容易になる．実証実験では，開発した API を用いてアプリケーションプログラムから複数の学術ネットワークを連携させたインタークラウド環境が動的に構築できることを示す．

A Study of a Token-based Authorization Mechanism for the International On-demand Network

Abstract: The NSI standard-based international on-demand network has been constructed over multiple R&E networks around the world. Such a network service requires a federation mechanism of authentication and authorization and an API, which allows application programs to control the network resources. In order to accelerate construction of international research platforms, we have developed a token authentication-based REST API mechanism. The mechanism enables to collaborate the GakuNin federation, the Shibboleth-based access management federation in Japan, and control the international on-demand network. Our experiment shows the feasibility of a constructing an international research platform using the developed API.

1. はじめに

学術研究の基盤として，大学や研究機関と高度実験装置，大規模スパコンセンター，学術・商用クラウドを接続した高性能，広帯域な学術ネットワークが世界的に配備されている．国立情報学研究所（NII）が運用している学術情報ネットワーク SINET5[1] は，図 1 に示すように日本国内の全都道府県のルータ間を 100Gbps 回線で相互接続された広帯域低遅延の国内ネットワークと，欧米アジアに接続する広帯域国際ネットワークを提供している．SINET5 に接続された複数拠点に対して L2 または L3 の安全かつ高性能な VPN（Virtual Private Network）を提供することができ，L2 VPN をオンデマンドで利用可能にする L2 オンデマンドサービス（L2oD）も提供している．欧州の GÉANT[2] や，米国の Internet2[3] も，同様の VPN サービスを提供している．これらの学術ネットワークから複数の商用クラウドプロバイダに直接接続されるようになったため，大学や研究機関と学術／商用クラウド間を VPN で接続し，安全かつ高性能なインタークラウド環境を構築することができるようになってきた．さらに，米国ではインタークラウド環境で研究のためのプラットフォームを構築する PRP（Pacific Research Platform）[4] プロジェクトが進められており，これを拡張した国際的な研究プラットフォームの構築も議論されている．このような国際的なインタークラウド環境を構築するには，異なる組織で管理されている学術ネットワーク間の連携が重要になる．各学術ネットワークがそれぞれどのように接続されているかを把握し，必要な端点間の VLAN などのネットワーク資源を相互に融通

1　国立情報学研究所
National Institute of Informatics
2　奈良先端科学技術大学院大学
Nara Institute of Science and Technology (NAIST)
a)　takefusa@nii.ac.jp

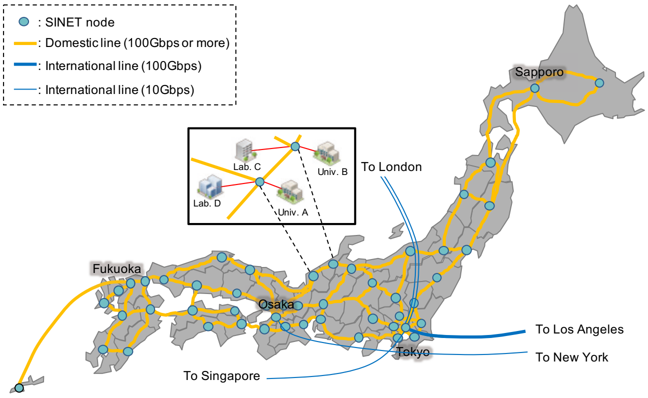

図 1　学術情報ネットワーク SINET5.

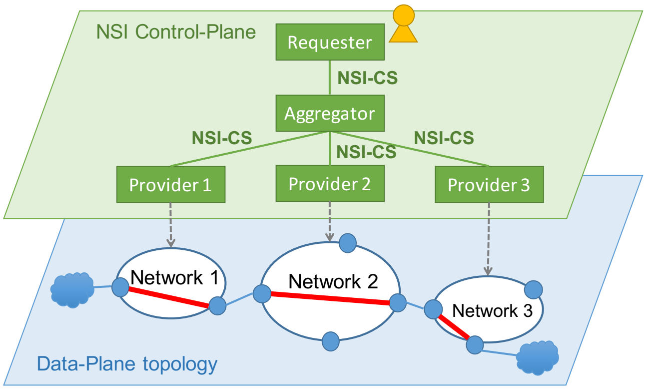

図 2　NSI アーキテクチャの概要.

しあう仕組みが必要となる.

Open Grid Forum（OGF）[5] の Network Services Interface（NSI）ワーキンググループでは，異なる組織により管理されているネットワークを共通の API で制御できるようにするため，事前予約ベースで VLAN 等のパスネットワークをサービスとして提供するためのコネクションサービス（CS）の標準仕様を規定した [6]．また，GLIF[7] の Automated GOLE ワーキンググループが主導して，NSI の CS v. 2 に準拠した国際的なオンデマンドネットワークテストベッド（NSI オンデマンドネットワーク）を構築し，複数の学術ネットワークを跨る VLAN を動的に構築する仕組みを提供している．この活動は，欧米アジアの複数組織により進められており [8]，SINET5 もそのネットワークの 1 つとして参加している [9].

NSI オンデマンドネットワークは試験的な運用であり，認証連携の仕組みが実装されていない．NSI オンデマンドネットワークのためのポータルシステムの実装の 1 つに MEICAN[10] があるが，ベーシック認証を前提としていて認証連携は実現できていない．よって，我々は NSI の認証認可に関する標準ドキュメント [11] に従い，SAML[12] ベースの認証機構をもつ Shibboleth[13] を用いた NSI ポータルサーバを開発した [9]．また，学術認証フェデレーション「学認（GakuNin）」[14], [15] に参加し，国内での認証連携を実現した．しかしながら，認証連携の仕組みの上でアプリケーションプログラムから制御する方法がないため，計算機等の他の資源との連携が容易ではなかった.

本研究では，国際的な研究プラットフォームの構築を加速させることを目指し，アプリケーションプログラムからの認証を可能にする，トークン認証ベースの REST API を開発した．これにより，従来はウェブからのマニュアル操作で VLAN を確保した後，アプリケーションプログラムから他の資源の確保やその上での環境構築を行っていたが，1 つのアプリケーションプログラムで全ての資源の操作が可能になる．本稿では，学認を用いた NSI ポータルサーバの詳細について述べるとともに，新たに開発したトークン認証ベース REST API について説明する．また，実証実験では開発した API を用いて，アプリケーションプログラムから異なる組織で管理されている学術ネットワークを用いたインタークラウド環境を動的に構築可能であることを示す.

2.　学認ベース NSI ポータルサーバの概要

我々は，標準インタフェースである NSI CS v. 2（NSI-CS）に準拠して NSI オンデマンドネットワークの VLAN を要求する NSI ポータルサーバを開発した [9]．NSI ポータルサーバは，Shibboleth と学術認証フェデレーション「学認（GakuNin）」を利用した認証認可機能と，NSI オンデマンドネットワークへの予約処理および予約の情報管理・監視のための GUI を提供している.

2.1　NSI CS v. 2 の概要

NSI-CS では，ネットワークサービスの要求者（Requester）と提供者（Provider）間のインタフェースを規定しており，必要に応じて仲介者（Aggregator）がサービス要求を仲介する．図 2 に NSI の想定するアーキテクチャを示す．Network 1，2，3 は異なる組織で管理されているネットワークであり，Provider 1，2，3 はそれぞれのネットワークの資源を管理するネットワークサービスエージェント（NSA）を表している．Requester は，各 Provider が管理しているネットワーク上の VLAN 等のサービスを要求する NSA である．Aggregator は，各ネットワークのトポロジを把握しておき，Requester の要求に応じて適切な Provider との調停を行う NSA である．各 NSA 間の通信は SOAP で行われており，WSDL（Web Services Description Language）で規定された Requester API と Provider API に基づいて行われる．Aggregator では，Requester に対して Provider API を，Provider に対して Requester API を提供している．NSI-CS は，非同期呼び出しを前提として設計されており，Requester，Provider とも Web サーバとして起動している．これは，複数のネットワークドメインが連携するために各リクエストの応答時間が長くなる可能性があるため，非同期的に呼び出すように設計されている．また，複

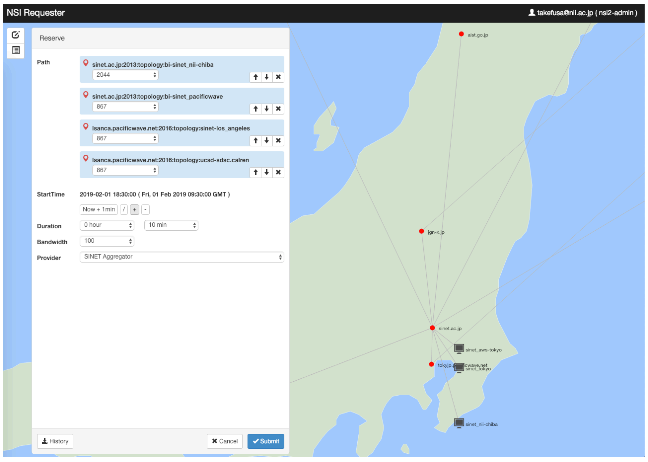

図 3　予約時の NSI ポータルキャプチャ画面.

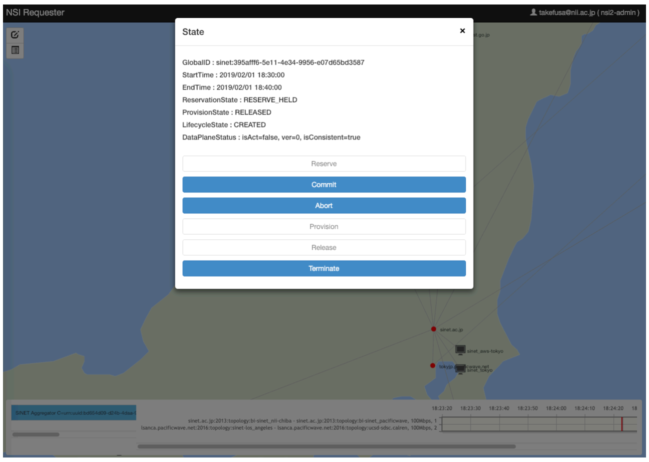

図 4　予約確定時の NSI ポータルキャプチャ画面.

数資源の同時確保に対応するため，二相コミットプロトコルとなっている.

2.2　NSI ポータルサーバ用 REST API

NSI-CS は SOAP ベースであり，REST API は規定されていない．末端の利用者に対する GUI を提供するため，NSI-CS に対応する REST インタフェースを独自に定義した．表 1 に NSI ポータルサーバ用 REST API を示す．NSI コマンドは，NSI-CS の Provider API で定義されたオペレーションに対応したものであり，予約，予約の確定／破棄，パスの配備／解放，予約の削除を行うことができる．また，NSI-CS の非同期呼び出しを踏襲しており，NSI コマンドの実行結果は query を用いて確認する.

NSI-CS では，個々の予約は Provider が払い出す Connection ID で識別することになっているが，Connection ID は予約が確定するまで払い出されず，NSI ポータルでの予約管理には利用できない．一方，Global Reservation ID（GlobalID）は NSI-CS では Requester 側が明示的に指定するパラメータで，それを Aggregator や Provider に引き渡す仕様になっている．NSI ポータルでは，GlobalID を識別子として利用することとし，NSI ポータルサーバ側で GlobalID を払い出して一意になるようにした.

NSI ポータルは，個々の予約要求の送信だけでなく，利用者が予約した資源全体の管理画面としての機能も提供する．そのため，ポーリングベースで query コマンドを実行する手法だと利用者数が増加するにつれサーバ側の負荷が高くなるため，Aggregator からコマンドに対する応答が送られた場合や定期的な query の結果を WebSocket で利用者のブラウザに送信する仕組みも実装している.

図 3，図 4，図 5 に，予約時，予約確定時およびパス配備後の NSI ポータルのキャプチャ画面を示す．図 3 では，どの端点間のパスを予約するかを指定している．図中の赤丸は各学術ネットワークを表しており，ここでは SINET の NII 千葉の端点から，Pacific Wave の UCSD の端点ま

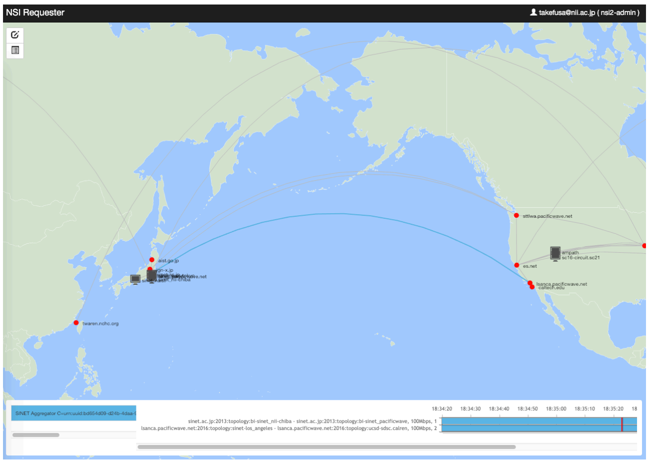

図 5　パス配備後の NSI ポータルキャプチャ画面.

でのパスと予約時刻，帯域，利用する Aggregator の情報を指定している．図 4 では，予約要求が成功後に commit を送信する際のキャプチャ画面を表す．ポップアップ画面の上部には，予約情報が記載されている．また，予約成功後の処理として commit，abort，terminate が可能であり，他のコマンドが誤って実行されないように実装されている．図 5 では，provision まで成功して予約時間にパス配備まで完了した状態をキャプチャした結果を示す．図 5 下部には予約情報のテーブルが表示されており，左側から GlobalID，予約されたパスの学術ネットワークおよび端点情報と予約時間を表すバーが示されている．また，地図上に予約したパスが GlobalID および予約情報テーブル内のバーの色と同じ水色の実線で表されている．これは，予約時刻でかつパス配備がされていない状態では破線で描画される．GlobalID 部分をクリックすると，図 4 で示されたような詳細な予約情報と実行可能な NSI コマンドのリストが表示できる.

2.3　SAML ベースの認証機構と学認連携

NSI の認証認可に関する標準ドキュメントでは，利用者

表 1　NSI ポータルサーバ用 REST API.

NSI コマンド	内容	HTTP コマンド	パス	ペイロード
reserve	予約／予約内容の修正	POST	/portal/reservations/	予約情報
commit / abort	予約の確定／破棄	PUT	/portal/reservations/{GlobalID}	NSI コマンド名を指定
provision / release	パスの配備／解放	PUT	/portal/reservations/{GlobalID}	NSI コマンド名を指定
terminate	予約の削除	DELETE	/portal/reservations/{GlobalID}	–
query	予約情報の取得	GET	/portal/reservations/{GlobalID}	–
–	トポロジの取得	GET	/portal/topology	–

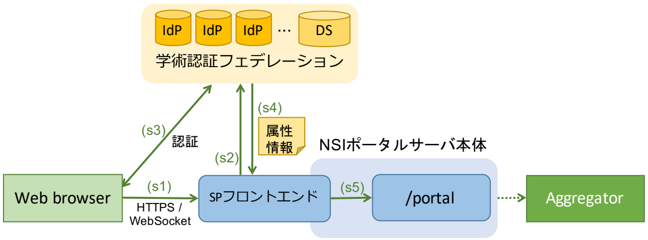

図 6　学認を用いた NSI ポータルサーバの概要.

の認証手法として HTTP ベーシック認証，X. 509（TLS)，OAuth，SAML が記述されている．また，NSA 間の認証情報のやり取りには，SAML のアサーションを利用することを推奨している．日本国内の大学および公的研究機関では，SAML および Shibboleth をベースとした学術認証フェデレーション「学認（GakuNin)」が認証連携の枠組みとして利用されているため，学認と連携する認証認可機構を実装した．

学認は，国内の大学，研究機関と NII が連携して運用されている認証連携基盤であり，Shibboleth を用いて構築されている．Shibboleth は，SAML や eduPerson 等の標準仕様を利用した認証・認可のための標準仕様策定とそのオープンソースを提供するプロジェクトである．学認により，各参加機関の利用者情報を管理する ID プロバイダ（IdP）の認証情報を使って，利用者は学認連携したサービスプロバイダ（SP）のサービスをシングルサインオンで利用することができる．NSI ポータルサーバも学認の SP として実装している．

図 6 に学認連携した NSI ポータルサーバの概要を示す．SP フロントエンドは Apache HTTPD，NSI ポータルサーバ本体は Apache Tomcat を用いて開発されている．通常は，ウェブブラウザから直接 NSI ポータルサーバにアクセスするのに対し，学認連携する場合は以下の (s1) から (s5) の手順でアクセスされる．

(s1)　ウェブブラウザから SP フロントエンドに対してリクエストが送付される．

(s2)　リクエストが学認基盤にリダイレクトされる．

(s3)　利用者の所属機関の IdP で認証する．

(s4)　認証に成功すると，アサーションを SP に送付する．アサーション情報には，アクセスした利用者の ePPN（eduPersonPrincipalName）と isMemberOf の属性情報が含まれている．

(s5)　NSI ポータルサーバ本体にリクエストとともにアサーションが送信される．

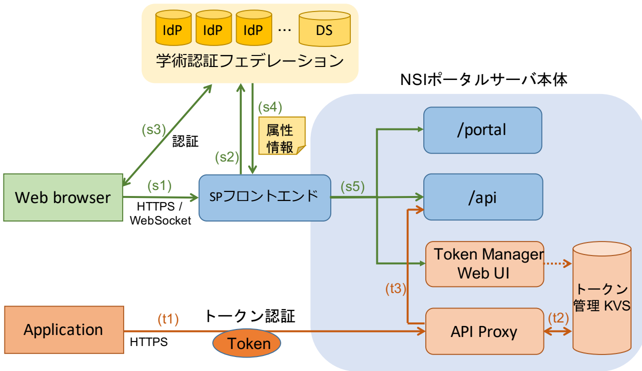

図 7　トークン認証ベース REST API を提供する NSI ポータルサーバの概要.

ePPN は個人を識別するための属性情報であり，isMemberOf は利用者のグループ情報，すなわち NSI オンデマンドネットワークでのロールを表す．isMemberOf は，学認のグループ管理システム「GakuNin mAP（member Attribute Provider)」を用いて，NSI ポータルサーバの管理者が各利用者に対してどのグループに所属させるか予め設定する．NSI ポータルサーバは，NSI の Requester の役割も担っており，これらの属性情報は SAML のアサーションで Aggregator にも引き継ぐ．今回の実装では，管理者，パワーユーザ，ユーザの 3 種類のグループを定義しており，グループにより GUI から可能な処理や表示結果を制御するようにした．具体的には，NSI ポータルサーバでユーザグループに属する利用者には自分の予約に関する処理とそれに関連する情報のみが表示されるようにし，管理者グループに属する利用者には全利用者の予約に関する制御や情報表示ができるようにした．また，NII が開発している Aggregator ではパワーユーザはユーザより多くの資源が予約できるような認可処理を行っている．

3.　トークン認証ベース REST API の開発

アプリケーションプログラムからの一元的な資源制御を可能にするため，トークン認証ベース REST API を開発した．トークン認証ベース REST API は，事前に Shibboleth 認証のトークン管理 Web UI でトークンを払い出した後，

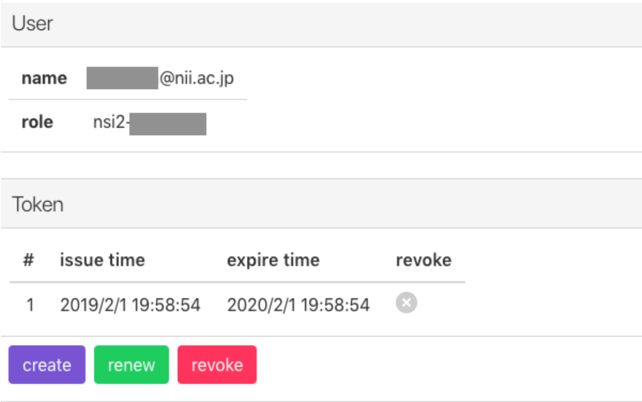

図 8　トークン管理マネージャのキャプチャ画面.

そのトークンを用いて呼び出される.

図 7 に，トークン認証ベース REST API を提供する NSI ポータルサーバの概要を示す．まず，2.3 節の (s1) から (s5) の手順でトークン管理マネージャにアクセスする．図 8 にトークン管理マネージャのキャプチャ画面を表す．トークン管理マネージャでは，create ボタンでアクセスした利用者用のトークンの払い出し，renew ボタンでトークンの有効期限の延長，revoke で払い出したトークンの破棄が行える．トークン管理マネージャは，トークン管理 KVS（Key Value Store）に生成したトークンの情報と学認の属性情報である ePPN と isMemberOf の値を保存しておく．トークン管理 KVS には，データを複数の鍵を用いて暗号化して管理する HashiCorp Valut[16] を利用しており，トークンは Vault が発行したしたものをそのまま利用している．トークン管理マネージャでは，各利用者が生成できる有効なトークンの数を制限している．

払い出したトークンを用いて，図 7 に示すようにアプリケーションプログラムから (t1) から (t3) の手順でトークン認証ベース REST API を利用する．

(t1) アプリケーションプログラムから API プロキシに対してトークンとともにリクエストが送付される．

(t2) API プロキシは，トークン管理 KVS にアクセスしてトークンが正しいことを確認するとともに，その利用者の学認の属性情報を取得する．

(t3) API プロキシが利用者の学認の属性情報とともにリクエストを送信する．

トークン認証ベース REST API は，基本的に表 1 の NSI ポータルサーバ用 REST API と同じであるが，パスの変更（/api）とヘッダ部分にトークンを追加する点のみが異なる．また，GlobalID は NSI ポータルサーバが自動的に付与していたが，トークン認証ベース API では利用者が一意に付ける必要がある．

4. 実証実験

実証実験では，本研究で開発した NSI オンデマンドネットワークのトークン認証ベース REST API を用いて，アプリケーションプログラムから異なる組織で管理されている学術ネットワークや計算資源を連携させることによって，インタークラウド環境を動的に構築可能であることを示す．具体的には，我々が現在までに構築している国際規模の SDN（Software Defined Networking）環境である PRAGMA-ENT（PRAGMA Experimental Network Testbed）を NSI による動的 VLAN を用いて拡張し，複数の学術ネットワークによるマルチパスネットワーク環境を構築する．

4.1 PRAGMA-ENT の概要

PRAGMA（Pacific Rim Applications and Grid Middleware Assembly）は主に環太平洋地域の大学及び研究所によって構成される国際共同研究コミュニティであり，e-Science を中心とした共同研究を継続的に長期間に渡って実施してきている．PRAGMA では，この e-Science における実証的研究のために，PRAGMA テストベッドと呼ばれる共有の計算機基盤を構築・運用している [17]．PRAGMA コミュニティに参画する各組織は，このような共有の計算機環境を構築するために，ある一定の計算機資源を相互に提供しあい，広域に分散した大規模な計算環境の構築を進めている．PRAGMA-ENT は，これら広域に分散する計算機資源を接続するネットワークをソフトウェアで柔軟に制御可能とし，様々な分散アプリケーションの実証実験を可能とするために，OpenFlow ベースの SDN テストベッド環境を構築するものである [18]．具体的には，現在までに JGN（Japan Gigabit Network）の RISE（Research Infrastructure for large-Scale network Experiments），Internet2，TWAREN（TaiWan Advanced Research and Education Network）等の学術ネットワークを活用し，5 カ国 10 組織を接続した広域 OpenFlow ネットワークを構築している．

しかしながら，現状の PRAGMA-ENT は各国の学術研究ネットワークの上で静的な拠点間 VLAN を手動で配備することによって構築しており，テストベッド上で実行されるネットワーク実験の実施スケジュールに応じて，構成変更などの手間を要している．インタークラウドにおける実証実験においては，個々の実証実験内容に応じて必要とする仮想ネットワークのトポロジや要件が異なり，また実験環境を必要とするのは評価作業を行う一時的な期間に限定される．そのため，手作業でネットワークの構成をその都度に変更する運用は非効率であり，実証実験を効率よく行えない．そのため，ソフトウェア制御により動的に複数のネットワークを連携させ，オンデマンドに実証実験基盤を配備可能な技術の確立が求められている．

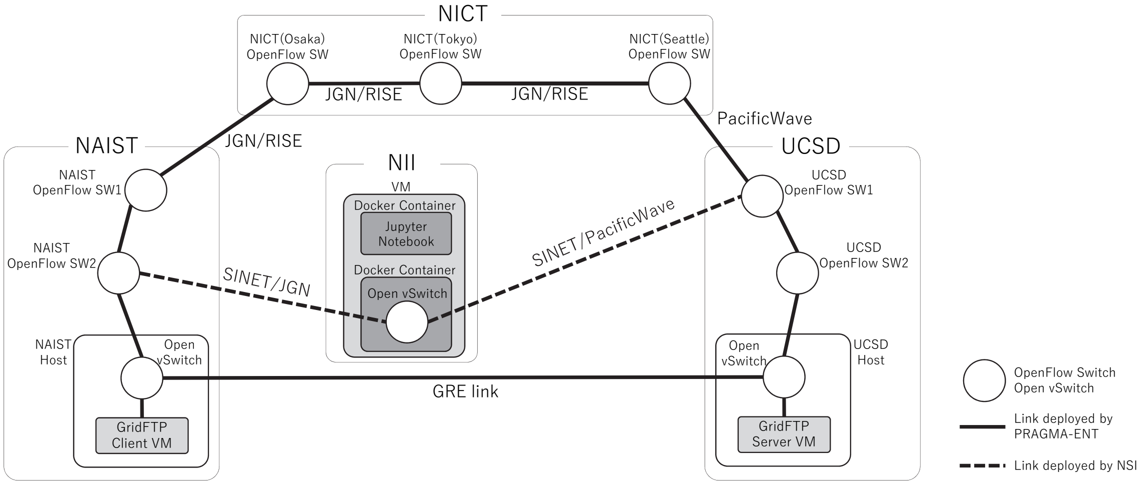

図 9　実証実験環境

4.2　実証実験の概要

今回の実証実験では，既存の PRAGMA-ENT のネットワークに対し，開発した NSI のトークン認証ベース REST API を用いて動的にネットワークを拡張することで，マルチパスネットワーク環境の構築を行う．そして，その際の API のパフォーマンスや，パスの確立に要する時間を評価する．

具体的な実験環境を図 **9** に示す．図中の実線で示されたリンクは既存の PRAGMA-ENT によって静的に配備されたリンクを表している．破線で表されたリンクが NSI によって動的に配備するリンクである．NII の計算機上の Docker コンテナに Jupyter Notebook 環境が配備されれており，この Jupyter Notebook から NSI オンデマンドネットワークの制御を行い，配備されたリンクを PRAGMA-ENT の OpenFlow 環境から制御可能とするために Open vSwitch へ接続する．この際の Open vSwitch も Docker コンテナ上に構築したものであり，動的に環境構築を行えるように設計している．NAIST と UCSD の計算機上に我々が開発した OpenFlow を用いたマルチパス通信に対応した GridFTP の Server と Client を配備しており [19]，NSI によって配備されたマルチパスを利用可能としている．

NSI による動的な VLAN は，表 1 に示す REST API を呼び出すことで配備可能である．図 **10** に NSI の REST API 呼び出しの擬似コードを示す．基本的には開始点となる STP（Service Termination Point）と終了点となる STP とそれぞれの VLAN ID をパラメータとして渡し，予約時間帯を指定し，reserve と commit，provision の各 REST API を順に呼び出すことでパスの予約とその配備を指示できる．一つのパスを張る際に，複数の STP を経由する必要がある場合は経由する STP 名のリストを stplist パラメータとして渡す必要がある．このように NSI の REST API を用いることで，Python プログラムから容易に VLAN 配備の要求を記述することが可能である．今回の実装では，実験に使用する GridFTP Server と Client は予め VM として配備したものを利用したが，商用・学術クラウド上の資源制御を可能にする学認クラウドオンデマンド構築サービス [20], [21] 等を利用すれば，インタークラウド環境を統合的に配備することが可能であり，非常に有益であると考えられる．

図 **11** に今回の実験環境における NSI の REST API 呼び出しに掛かる処理時間の計測結果を示す．パスの利用を開始するためには，reserve，commit，provision の 3 つの API 呼び出しを順次完了させる必要があるため，この 3 つの API 呼び出しに掛かる処理時間を積み上げグラフとしてまとめている．また，経由地がなく VLAN 予約一つで直接パスが作れる経路と経由地があるパスによって処理時間が異なるため，別グラフとして集計している．計測結果より，直接パスが作れる経路ではおよそ 3 秒程度で reserve，commit，provision が完了していることが分かる．一方で，経由地があるパスの場合は 2 つの学術ネットワークの Provider での予約処理が行われるため，処理時間が長くなっていることが分かる．

これら API 呼び出しに掛かる時間は，各パスの予約ごとに掛かる処理時間であり，実証実験で必要となるパスが複数ある場合はその分処理時間を要することになる．ただし，異なるドメイン間の VLAN 配備が必要となるのは，実証実験の開始時および環境の変更時のみであるため，高頻度でこれらの API が呼び出される状況は想定されない．したがって，数秒程度の処理時間は実用的には十分許容範囲内であると考えられる．

一方で，予約開始時刻に達してから実際にパスが利用可能となるまでにも，少し時間を要することが分かっている．これは NSI の Provider が予約開始時刻になってから実際に VLAN の設定を各スイッチに投入し，その処理が完了するまでにある程度時間を要するのと，アプリケーションレベルでパスの疎通確認が取れて実際に利用可能となるま

```
globalid = "sinet:{}".format(str(uuid.uuid4()))
sourcestp = "Name of source STP"
sourcevlan = "Source VLAN ID"
deststp = "Name of destination STP"
destvlan = "Destination VLAN ID"
stplist = [List of names of way-stop STPs]
bandwidth = "Required bandwidth"
starttime = 'YYYY-MM-DDThh:mm:ss+ssss'
endtime = 'YYYY-MM-DDThh:mm:ss+ssss'
providernsa = 'SINET Aggregator'
headers = {
    'Content-Type': 'application/json',
    'Authorization': 'Bearer {}'.format(token)}

## Reserve
url = 'https://ENDPOINT-URL/reservations'
payload = json.dumps({
    'globalid': globalid,
    'sourcestp': sourcestp,
    'sourcevlan': sourcevlan,
    'deststp': deststp,
    'destvlan': destvlan,
    'bandwidth': bandwidth,
    'starttime': starttime,
    'endtime': endtime,
    'providernsa': providernsa,
    'stplist': stplist})
requests.post(url, headers=headers, data=payload)

## Commit
url = 'https://ENDPOINT-URL/reservations/'
  + urllib.parse.quote(globalid)
payload = json.dumps({'operation': 'commit'})
requests.put(url, headers=headers, data=payload)

## Provision
payload = json.dumps({'operation': 'provision'})
requests.put(url, headers=headers, data=payload)
```

図 10　NSI REST API 呼び出しの擬似コード

でに時間を要することから生じる．特に今回の実証実験で用いた OpenFlow コントローラでは 60 秒ごとに各スイッチ間の疎通を LLDP プロトコルにより確認し，トポロジを更新する実装であったため，アプリケーションレベルでパスが利用可能となるまでには予約開始時間後，約 40〜110 秒程度の範囲で時間を要することとなった．これは，NSI の query API を利用すればパスの配備状況が取得できるため，NSI 上でパスがアクティベートされたことが検知できたら，OpenFlow コントローラに通知し，トポロジの更新を直ぐに実施するなどの実証上の工夫を施すことで，改善の余地があると考えられる．

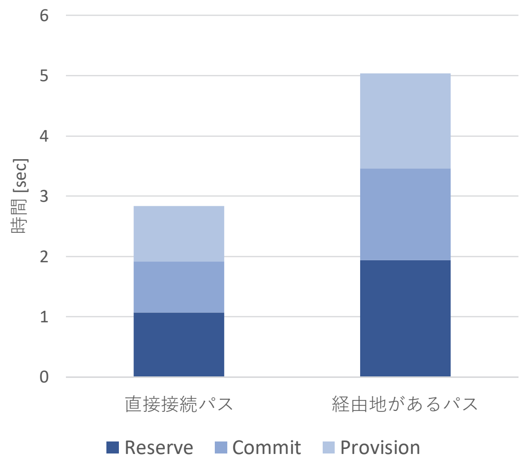

図 11　NSI の REST API の処理時間

5. おわりに

本研究では，国際的な研究プラットフォームの構築を加速させることを目指し，アプリケーションプログラムからの認証を可能にする，トークン認証ベースの REST API を開発した．これにより，アプリケーションプログラムから国際的なオンデマンドネットワークの制御が可能になり，計算機等の他の資源との連携も容易になる．また，実証実験では開発したトークン認証ベース REST API を用いて，アプリケーションプログラムから異なる組織で管理されている学術ネットワークや計算資源を連携させることによって，インタークラウド環境を動的に構築可能であることを示した．今後は，商用・学術クラウド上の資源制御を可能にする学認クラウドオンデマンド構築サービスと連携させ，高度なインタークラウド実験環境の構築を行う．

謝辞　本研究成果の一部は，平成 30 年度国立情報学研究所公募型共同研究および，JSPS 科研費 18K11326 の助成を受けて実施されたものである．本研究の実装にご協力いただいた数理技研の小泉敦延様に深く感謝いたします．

参考文献

[1] 学術情報ネットワーク SINET5, https://www.sinet.ad.jp/ (accessed on 01-02-2019).
[2] GÉANT, https://www.geant.org/ (accessed on 01-02-2019).
[3] Internet2, https://www.internet2.edu/ (accessed on 01-02-2019).
[4] Pacific Research Platform (PRP), https://prp.ucsd.edu/ (accessed on 01-02-2019).
[5] Open Grid Forum (OGF): http://www.ogf.org/.
[6] Roberts, G., Kudoh, T., Monga, I., Sobieski, J., MacAuley, J. and Guok, C.: NSI Connection Sevice v2.0, *GFD. 212*, pp. 1–119 (2014).
[7] GLIF: Global Lambda Integrated Facility, http://www.glif.is/.
[8] 竹房あつ子，工藤知宏，高野了成，中田秀基，大久保克彦，岡崎史裕：NSI コネクションサービスプロトコル ver. 2 の参照実装の開発，信学技報 NS2013-202，pp. 149–154

(2014).

[9] 竹房あつ子，栗本　崇，合田憲人：SINET5 と NSI ネットワークテストベッドを用いた国際間オンデマンドネットワークの構築，信学技報 NS2016-165，pp. 47–52 (2017).

[10] MEICAN, `https://github.com/ufrgs-hyman/meican` (accessed on 01-02-2019).

[11] Trompert, H. and MacAuley, J.: NSI Authentication and Authorization, *GFD. 232*, pp. 1–23 (2017).

[12] S. Cantor et al.: Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, `https://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf` (accessed on 01-02-2019).

[13] Shibboleth Consortium - Privacy Preserving Identity Management, `https://shibboleth.net/` (accessed on 01-02-2019).

[14] 学術認証フェデレーション，`https://www.gakunin.jp/` (accessed on 01-02-2019).

[15] 西村　健，中村素典，山地一禎，佐藤周行，大谷　誠，岡部寿男，曽根原登：多様なポリシーを反映可能な認証フェデレーション機構の実現，電子情報通信学会論文誌，pp. 1400–1412 (2013).

[16] HashiCorp Vault, `https://www.vaultproject.io/` (accessed on 01-02-2019).

[17] Tanaka, Y., Yamamoto, N., Takano, R., Ota, A., Papadopoulos, P., Williams, N., Zheng, C., Huang, W., Pan, Y.-L., Wu, C.-H., Yu, H.-E., Shiao, J. S., Ichikawa, K., Tada, T., Date, S. and Shimojo, S.: *Cloud Computing and Big Data*, Vol. 23, chapter Building secure and transparent inter-cloud infrastructure for scientific applications, pp. 35–52, IOS Press (2013).

[18] Ichikawa, K., U-Chupala, P., Huang, C., Nakasan, C., Liu, T.-L., Chang, J.-Y., Ku, L.-C., Tsai, W.-F., Haga, J., Yamanaka, H., Kawai, E., Kido, Y., Date, S., Shimojo, S., Papadopoulos, P., Tsugawa, M., Collins, M., Jeong, K., Figueiredo, R. and Fortes, J.: PRAGMA-ENT: An International SDN testbed for cyberinfrastructure in the Pacific Rim, *Concurrency and Computation: Practice and Experience*, Vol. 29, No. 13, p. e4138 (online), DOI: 10.1002/cpe.4138 (2017).

[19] Huang, C., Nakasan, C., Ichikawa, K., Watashiba, Y. and Iida, H.: A Multipath OpenFlow Controller for Multiple TCP Stream Applications, *Journal of Information Processing*, Vol. 25, pp. 924–933 (online), DOI: 10.2197/ipsjjip.25.924 (2017).

[20] 学認クラウドオンデマンド構築サービス, `https://cloud.gakunin.jp/ocs/` (accessed on 01-02-2019).

[21] Takefusa, A., Yokoyama, S., Masatani, Y., Tanjo, T., Saga, K., Nagaku, M. and Aida, K.: Virtual Cloud Service System for Building Effective Inter-Cloud Applications, *Proc. IEEE CloudCom2017*, pp. 296–303 (2017).