

統一コンフィグレーション実現のための アプリケーションサーバ試験方式の提案

吉田 敦† 三好 優† 大塚 祥広†

† 日本電信電話株式会社 NTT ネットワークサービスシステム研究所
〒 180-8585 東京都武蔵野市緑町 3-9-11
E-mail: †{yoshida.atsushi,miyoshi.yu,otsuka.yoshihiro}@lab.ntt.co.jp

あらまし 論理的多層構造を持つ IP 網や IP をベースとした NGN では、本来統一的な運用が望ましいにもかかわらず、複数のツールを組み合わせた複雑な運用をせざるを得ない状況にある。我々は、上位レイヤであるサービス・アプリケーション管理から下位のネットワーク機器管理レイヤに至るまでの運用をオペレータが一貫して行なうための基盤技術として、現在開発しているコンフィグレーション支援技術により、分割されている様々なオペレーションレイヤにおける試験機能の統合による運用効率向上を目指している。本稿では、NGN におけるネットワーク制御サーバ系の運用方法の一ソリューションとして、オペレータがパラメータを変更するなどの簡易な作業のみで Web サービスのアプリケーションに対する試験を行なうための機能要件を整理し、オペレータの試験業務の負荷軽減を実現する方式を提案する。また、提案する試験方式を実現するための要素技術の動作確認を検証したので報告する。

キーワード NGN, NGN オペレーション, Web サービス

Proposal of Application Servers Verification Methods for Unified Configuration Management

Atsushi YOSHIDA†, Yu MIYOSHI†, and Yoshihiro OTSUKA†

† NTT Network Service Systems Laboratories, NTT Corporation
9-11, Midori-Cho 3-Chome Musashino-Shi, Tokyo 180-8585 Japan
E-mail: †{yoshida.atsushi,miyoshi.yu,otsuka.yoshihiro}@lab.ntt.co.jp

Abstract The IP network and Next Generation Network (NGN) have logically a multi-layer configuration. Network carriers must operate the networks with various tools and complex procedures. We try to establish methods for realizing that operators can consistently process multi-layer operation from upper layer which is service or application management to lower layer which is network element management. Now, we try to use the configuration support tool which is developed by NTT laboratories. At first, we make ourselves clear about the issues of application server test functions. And we demonstrate the effectiveness and impact of realizing test functions for the application server.

Key words NGN, NGN Operation, Web service

1. はじめに

新しい通信サービスに対応するため、BT を始め世界各国の通信キャリアが既存の公衆電話網 (PSTN: Public Switched Telephone Network) に代わる次世代ネットワーク (NGN: Next Generation Network) の導入を目指し、ITU-T, ETSI (Euro-

pean Telecommunications Standards Institute: 欧州電気通信標準化機構) TISPAN (Telecom & Internet converged Services for Advanced Networks) などで NGN の標準化議論が活発に行なわれている [1] [2] [3].

NGN は、IP 技術をベースにしたネットワーク上に既存電話網の互換サービスさらにはマルチメディ-

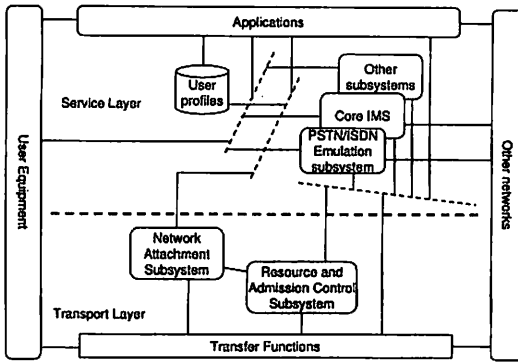


図 1 TISPAN NGN Architecture

アサービスなど高度なサービスの多重を可能にする統合ネットワークである。TISPAN で検討が進められている NGN のアーキテクチャ (図 1) は、サービス層とトランスポート層による階層構造となっており、従来のネットワーク機器をトランスポート層として定義するだけでなく、上位のサービスを提供するアプリケーションや各種サービスを提供するコアサブシステム群を、サービス層に配置することで通信インフラストラクチャの一部として定義している。このサービス層の各コアサブシステム群の働きにより、移動・固定融合 (FMC: Fixed Mobile Convergence) やトリプルプレイ、サードパーティによるサービス展開といったビジネス基盤を構築することが可能となる。

我々 NTT 研究所においても、NGN によって既存のインターネットの欠点を克服し、安心・安全を提供することを目標の一つに掲げ、トランスポート層における大容量・経済的な伝送系コアネットワークから構成されるネットワーク構成技術と共に、サービス層における高機能・高付加価値なサービスを提供するネットワークサービス制御技術の開発を行っている。

2. NGN オペレーション

NGN を成功させるためには、アーキテクチャや機器など構成要素に関わる技術の研究開発のほかに、その管理・運用手法についても様々な技術開発が必要である。安心・安全が求められるネットワークサービスの運用には、サービスを多重したネットワーク上で個別サービスごとに設定された品質遵守が必要であり、NGN の運用管理は従来以上の複雑な手順と技術が求められる。NGN では、従来の IP 網の運用技術だけでなく、サービス層に配置されるネットワークサービス制御系のサーバの運用技術が求められる。

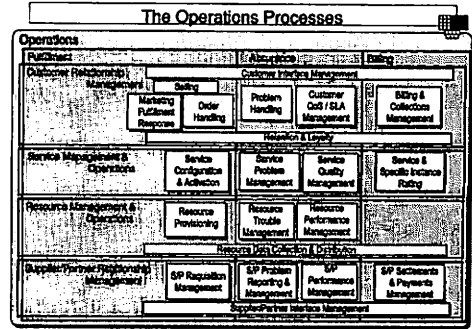


図 2 eTOM Operations Processes

TISPAN の WG8 など NGN マネージメントについて議論が進められているが [4], OSS (Operations Support System) の構成やビジョンの策定を進めている段階であり、NGN のアーキテクチャ構成の特徴を踏まえた運用手法に関する議論にまでは至っていない。

一方、次世代のオペレーションに関する技術動向として、TeleManagement Forum (TMF) が進めている次世代 OSS のためのアーキテクチャ・フレームワークが NGOSS である [5]。NGOSS は NGN に対するオペレーションに対しても適用可能な、次世代オペレーションの有望な技術であると期待されている。

NGOSS には、目的に応じて 4 つのフレームワークが準備されている。その一つである eTOM (Enhanced Telecom Operations Map) [6] では、テレコムに関する業務やビジネスプロセス全般が分類されており、既存・将来の OSS の機能の位置づけがマップされるようになっている (図 2)。eTOM では特にネットワーク管理層およびサービス管理層のビジネスプロセス細分化の策定が進められており、ネットワーク管理機能やサービス管理機能の位置づけと特性が詳細にまとめられている。

eTOM や TAM (Telecommunication Application Map) といったこれらのフレームワークに象徴されるように、現在の OSS や運用ツール機能はネットワーク管理層、サービス管理層といったオペレーション機能による区分によって棲み分けがされている。つまり、同じサービスの運用でも、異なる対象機能であれば異なるツールや OSS が必要であったり、また同一の対象であっても、例えば Ping や Telnet を使った機器に対する従来の試験方法と、その機器上で動作するアプリケーションの試験方法が全く異なったツールと方法が必要であるように、複数のツールや複雑な手順が必要であり、一貫した運用を行いにくいことを示している。

そこで我々は、NGN オペレーションに対する取り組みとして、まず、同一の対象に対し Ping や Telnet による機器へのプリミティブな試験からサービスを提供するアプリケーションに対する高度な試験までを一貫して行なう、統一型オペレーション方法とその支援技術について検討を進める。

まず、NGN におけるサービス層の運用に着目し、トランスポート層の機器からサービス層に対する機器についても統一された操作を行なうことでオペレータのユーザビリティを高めるオペレーション支援技術の検討を進める。

3. アプリケーションサーバ試験方式の提案

本稿では、NGN においてネットワークインフラとして新たに加わるネットワークサーバ制御系の管理方法の確立を行なうとともに、既存のキャリア網管理からのスムーズな運用方法のマイグレーションと、NGN 導入によるオペレーションコスト増加のインパクトを抑えることを目的とする。NGN におけるネットワーク制御系サーバを想定し、機器の通信機能から機器で動作する上位アプリケーションまでを一貫して管理する、NGN サーバの管理方法について述べる。

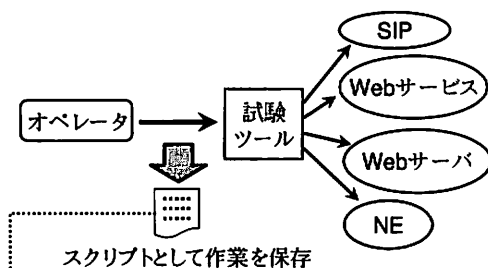
次に、本提案で利用する、トランスポート層の機器 (NE: Network Element) に対する管理運用技術として開発を進めているコンフィグレーション支援技術について説明する。

3.1 アプリケーションサーバ試験方式

NGN 上に配備されるサービスに使用される技術には、Web サービスを実現する際に使用される UDDI・SOAP・WSDL、VoIP に使用される SIP など、テキスト形式のメッセージを送受信するものが多い。各技術において送受信されるメッセージは、各技術の実装が提供するライブラリによって隠蔽され、ユーザの目には直接見えない状態となっている。これは、サービスを利用もしくは構築するユーザ側の立場としては、生産性を向上するために必要な手法である。しかし、プラットフォームを構築し、サービスの品質を保証しなくてはならないキャリア側の立場からすると、サービスの監視・制御が行いにくい状況とも言える。

この場合、通信キャリアは、サービスの監視・制御を行うために、専用のモニタリングソフトを用いて、サーバ・クライアント間において送受信されるテキスト形式のメッセージを表示・解析する必要がある。したがって、現在は複数のツールを使用する必要がある。これらの検証環境を一元化することができれば、円滑なワークフローが構築され、検証業務の効率化を進めることが可能であると考えられる。

① NGN 上のサービスにおける統一運用環境を実現



② 自動化による作業の効率化

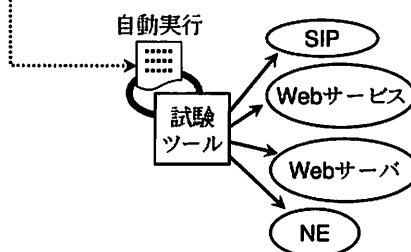


図 3 Proposal Method

今回我々は、使用されるメッセージがテキスト形式で記述されている点に着目し、テキスト処理を行うソフトウェアを使用して、サービスの統一的な監視・制御を実現する手法を提案する (図 3)。テキスト形式のメッセージを使用する SOAP・WSDL・SIP などの技術において、アプリケーションが送受信するメッセージの内容をオペレータが視認できる形式で開示し、メッセージの送受信の制御を支援することで、アプリケーションに対する管理業務をオペレータとの親和性を保ちつつ実現することが可能である。

具体的には、現在、NE の設定や検証などの運用業務にて使用されている CLI (Command Line Interface) ベースの支援技術を用いて、サービスの運用に利用する手法を提案する。これにより、オペレータは NE の運用の際と同様の操作で、サービスの運用を実施することが可能となる。オペレータにとって統一的な運用環境を構築し、サービス単位での IP ネットワーク管理を実現する。運用環境が統一されることによって運用業務が単一のソフトウェアの制御下に収まるため、自動化などの効率化が実現しやすくなり、また、自動化が実現した際の適用範囲が広がる。オペレータは、複数のツールや専用のクライアントを構築・操作する必要はなく、NGN における運用業務の効率化を実現する。

3.2 コンフィグレーション支援技術

現在我々は、ルータやスイッチなどの NE に対し、

オペレータの既存の技術や知識を継承しつつ、設定や検証などの運用業務を高度化し、オペレーション負荷を軽減するコンフィグレーションを支援する技術の研究開発を行っている(図4)[7]。

現在、ポリシー設定や利便性の高いGUIを持つCOTS(Commercial Off The Shelf)製品など、高度な機能を持つ運用技術が存在するにもかかわらず、ネットワーク機器の検証業務や設定やバックアップ、情報収集などのコンフィグレーション管理業務は、従来のCLI上での処理が未だ主流となっている。これには、一度作りこんだ高度な機能がネットワーク仕様の更新スピードについていけないことなど様々な理由が考えられるが、大きな理由として、既に多くのオペレータの知識や経験が従来のCLIに蓄積されていることが考えられる。そこで我々のコンフィグレーション支援技術は、オペレータとの親和性が高いCLI上での操作を継承し高度化するアプローチによってオペレータの業務負荷を軽減する手法をとっている。開発したコンフィグレーション支援ツールは、オペレータの動作処理をスクリプト形式で記録し、記録の再利用の再生実行、タイマー実行する機能を備えるため、オペレータは一度通常のCLI上の作業を行なうだけでその後の同様の操作を自動実行することができ、ルーチンワークによる業務負荷の大幅な軽減を可能とする。また、スクリプトをオペレータにとって見やすい処理記録フォーマットとすることで、スクリプトに記録されているコマンドや、オペレータがその場で書き換えるコマンド実行の条件を変更することを容易とした。これにより、オペレータが柔軟にパラメータ変更などを行ないつつ現場で試験項目を策定し、実行することが可能になった。特に検証環境においてはプログラミングなど開発により生じるタイムラグを削減し、現場で柔軟に試験を実施することが可能となると考えられる。

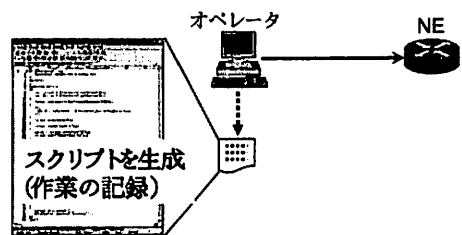
4. 検証

4.1 実験概要

これまで述べてきた提案の実現可能性を確認するため、実験を行う。今回は、対象サービスとしてWebサービスを選択し、また、試験対象として、Webサービスを提供するサーバを選択した。

Webサービスの利用形態として、Webサービスを提供するサーバ、Webサービスを組み合わせてサービスを提供するウェブサイト、サービスを利用するクライアントによって構成されるのが一般的である。Webサービスを提供するサーバには、WSDLを提供する機能と、SOAPメッセージの送受信によりWebサービスを提供する機能が必要である。以上に示した2つの機能の監視・制御が、サーバの検証には必要不可欠であると考えられる。

① CLIによる設定をスクリプトとして保存



② スクリプトを利用して、自動実行

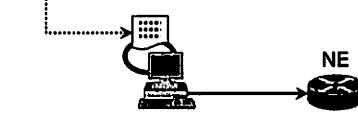


図4 Configuration Support Tool

ここで、Webサービスを提供するサーバを検証する際に必要な要素を整理する。Webサービスは、複数のリソースから構成されているためサービスに障害が発生した際には当該サービスを構成するリソースの状態を順番に確認し、障害が発生している原因箇所を特定する必要がある。サーバの検証を実施する際に、状態を確認する必要がある要素を以下に示す。

- (1) IPパケットの到達性
- (2) HTTPデーモン
- (3) Webサービスを構築するプラットフォーム
- (4) WSDL
- (5) Webサービス

現状でのサーバの検証手法では、以上に示した要素を確認する際にそれぞれ別のツールを使用する必要がある。また、手作業で実施しなくてはならない箇所が存在し、自動化の必要が十分にあると考えられる。

4.2 実験

本手法で提案するWebサービスを提供するサーバの試験手順について説明を行う。まず、サーバに対してpingコマンドを使用して、IPパケットの到達を確認する。次に、指示されたHTTPリクエストを送信することにより、HTTPデーモンの確認・Webサービスを構築するプラットフォームの確認・WSDLの取得を行う。取得したWSDLに対して文法チェックを行い、文法が正しいことを確認した後に、対象Webサービスが提供するメソッドを提示する。提示されたメソッドを適切な順番で実行し、その応答を確認する。一度、適切な検証が実施された後は、その際に使用されたXMLメッセージをスクリプトとして保存し、オペレータがそのスクリプトを変更・再送

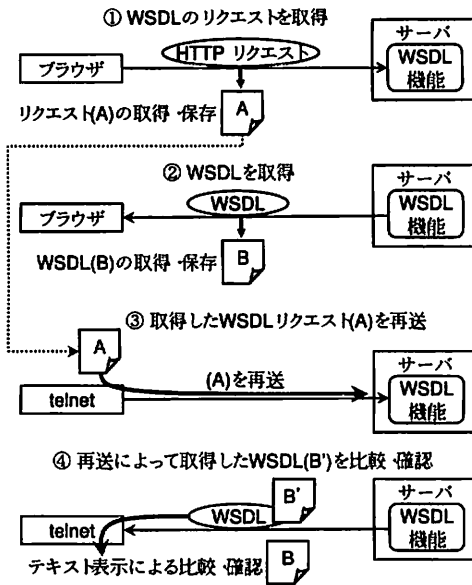


図 5 WSDL の確認実験

することにより、目標のサーバにおける試験の自動実行ができるようにする。今回は、以上の手順の実現可能性を確認するために、実験を行う。

実験は telnet を使用して、Web サービスを提供するサーバに対して、WSDL と Web サービスの提供を要求する。WSDL を取得する際には、ブラウザが作成した HTTP リクエストを取得し、telnet を使用して再送することによって WSDL の取得を行う。この実験により、サーバにおける WSDL 提供機能の試験の実現可能性を確認する (図 5)。

Web サービスを利用するには、専用のクライアントが作成した SOAP メッセージを取得し、telnet を使用して再送することによって Web サービスを利用する。この実験により、サーバにおける Web サービス提供機能の試験の実現可能性を確認する (図 6)。また、Web サービスに対する実験を行う際には、SOAP メッセージ内のテキストを変更し、パラメータを変更した Web サービスの検証をテキスト処理のみで実現できることを確認する。

サーバが提供する WSDL と Web サービスの 2 つの機能について、以下に示す視点で実験を行い、提案手法の有効性について確認を行った。

- 確認項目 1: 機能を監視可能であるか (Monitoring)
- 確認項目 2: 様々な試験が実行可能であるか (Experiment)
- 確認項目 3: 監視・試験機能の自動化が可能であるか (Automation)

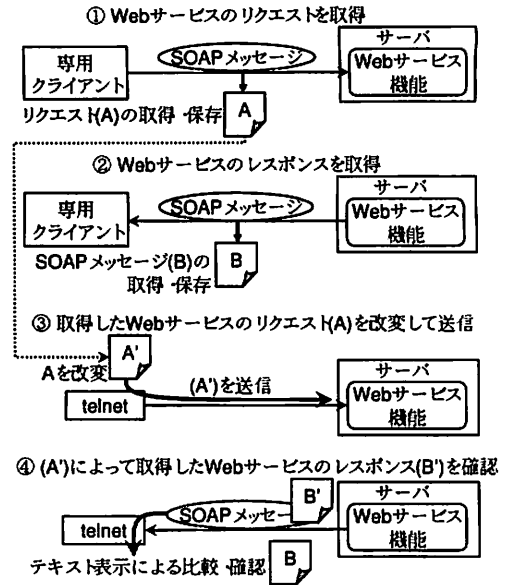


図 6 Web サービスの確認実験

各実験の確認項目について説明する。

Monitoring 専用のモニタリングソフトを使用せずに、汎用的な環境でサーバの監視が実現可能かを確認する。

Experiment 専用のクライアントソフトを使用せずに、汎用的な環境でサーバの試験が実現可能かを確認する。単に試験が実施できるだけでなく、テキストを書き換えるだけでパラメータを変更した Web サービスの試験が実現可能であれば、サーバの統一的な試験環境の構築へとつながる。

オペレータによるテキスト処理や自動化が可能か、さらに実験結果から考察を行う。

Automation 上記の機能の自動化が可能であるかを、実験結果から確認する。また、自動化を行うために必要な要件の整理を合わせて行う。

4.3 実験結果

実験の結果を、表 1 に示す。Web サービスを提供するサーバにおける WSDL 機能の試験を実施するため、ブラウザが生成した HTTP リクエストを取得し、再送を行った。クライアント・サーバ間で送受信されたメッセージを確認し、その結果、正常に WSDL を取得できることを確認した。

Web サービスの試験を実施するため、専用のクラ

表 1 実験結果のまとめ

	再送	結果の同一性
WSDL の確認実験	Y	Y
Web サービスの確認実験	Y	Y

クライアントソフトが生成した SOAP メッセージを取得し、再送を行った。クライアント・サーバ間で受信されたメッセージを確認し、その結果、サーバから正常な結果が返信されることを確認した。その際、Java 言語を使用して実装されたクライアントソフトウェアを使用する場合と比較して、環境変数の設定や Java ソースのコンパイルなどの作業が必要なく、より汎用的なサーバ試験環境を構築することが可能であることを確認した。今回は手動で実験を行ったため、一度のリクエストの際に複数回のバケットが生成されたが、Web サービスの実現には影響はなかった。

リクエストのパラメータ変更を実施しながらの試験機能については、SOAP メッセージ内においてテキスト表記されているパラメータ値と HTTP ヘッダを書き換えることにより、パラメータを変更したリクエストを送信可能であることを確認した。その際のサーバからの返信も正常であった。

4.4 考 察

実験の結果に対する考察を以下に述べる。WSDL の Monitoring については、通常の HTTP リクエストであるため、定期的な監視が可能である。WSDL の Experiment については、WSDL は Web サービスの仕様に変更が発生するまで同一であるため、パラメータを変更しながらの再送処理については必要はないと判断した。

上記 2 つの機能の Automation については、WSDL の存在する URL さえ指定すれば、スクリプトなどによる自動化が可能であると判断した。WSDL 取得機能を自動化を行い、さらに取得した WSDL をスクリプト内に保存することによって、サーバの自動監視や障害復旧時の自動試験、および Web サービスの自動実行機能に利用することが可能であると考えられる。

Web サービスの Monitoring については、一度適切な SOAP メッセージを取得できれば、監視機能を構築することは可能である。今後は、最初に正常な SOAP メッセージを作成する作業の効率化が課題であると考えられる。Web サービスの Experiment については、SOAP メッセージと HTTP ヘッダを書き換えることにより、パラメータを変更した Web サービスの試験が実施できることを確認した。HTTP ヘッダの書き換えは機械的なものであり、テキスト処理を行うソフトウェアで十分に自動化が可能であると思

われる。また、書き換えられるパラメータの値の種類は、SOAP メッセージで適切に指示されるために、入力正当性を自動で確認することが可能であると考えられる。

上記 2 つの機能の Automation については、一度 Web サービスのリクエストを取得した後は、テキストに対する処理のみで様々な試験が自動で実現可能であると考えられる。その際、変更するパラメータの範囲をあらかじめオペレータが入力しておく必要があると考えられる。

結論として、NE に対する監視・制御手法と同様の手法で、Web サービスを提供するサーバに対する監視・試験が実施可能であると考えられる。以上の考察のまとめを表 2 に示した。

5. おわりに

本稿で提案した手法の実現により、IP ネットワーク上に配置された Web サービスを提供するサーバを対象とした試験作業を行う際に、オペレータに必要な手間と時間を削減することが可能であると考えられる。具体的には、サーバに対する試験作業を行う際に、以下に挙げるような効果が期待できる。

- テキストを書き換えるといった簡単な操作で、通信機能に対する制御および Web サービスの監視・試験を実施し、Web サービスを提供するサーバに対する運用環境を統一する

- 作業の自動化が容易である
- 専用のクライアントを構築する必要がない

今後は、パラメータの自動変更を伴った試験機能の自動化を推進する。さらに、Web サービスを提供するサーバ以外のサーバ、およびクライアントに対する監視・試験機能を追加し、より汎用的な検証環境を目指す。クライアントを対象とした試験機能を実現することにより、サーバ実機に負担をかけることなく試験を実施することが可能である。

文 献

- [1] 黒川 章, 東 勲, “ETSI TISPAN の NGN 標準化動向,” NTT 技術ジャーナル Vol.18, No.4, Apr. 2006.
- [2] ITU-T Y.2001, “General overview of NGN”
- [3] ETSI TR 180 001, “NGN Release 1 Definition”
- [4] ETSI TS 188 001, “Network Management Operations Support Systems Architecture”
- [5] TMF-TMF051, “TM Forum: New Generation Operational Support System (NGOSS) PART I (Business Case),” TMF051, 2001.
- [6] GB921, “The Business Process Framework”
- [7] 三好 優, 木村 辰幸, “インタフェース船舳・調査技術,” NTT 技術ジャーナル Vol.17, No.8, Aug. 2005.

表 2 実験の確認項目のまとめ

	確認項目 1: Monitoring	確認項目 2: Experiment	確認項目 3: Automation
WSDL	Y	-	Y
Web サービス	Y	Y	Y