社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－DPS－136 (8)
2008－GN－ 69 (8)
2008－EIP－ 41 (8)
2008／9／25

リソース指向型異種ネットワーク協調管理機構の設計

川原圭博[†1] 瀬崎薫[†2] 浅見徹[†1]

ネットワークサービスを継続的に提供するためには，サービスを提供するコンピュータおよびネットワーク機器をモニタし，必要に応じて制御を行うネットワーク管理機構が必要不可欠である．ところが現在では，一つのネットワークサービスの構成要素が複数の事業者によって構成されることも稀ではなく，運用情報の交換自体が難しくなっている．一般ユーザにとっては，「なぜかサービスが使えない」ことしかわからず，誰に解決を求めれば良いかもわからないといった状況が発生する．本稿では，Web サービス連携技術である REST アーキテクチャスタイルに基づいた管理 API について示す．これにより HTTP を介したリソース指向の管理情報の交換と制御が可能になり，ドメインをまたがったサービスの構成要素の管理情報にプログラムからアクセスしやすくなると期待できる．

A Design of Resource Oriented Heteronegeous Network Management Mechanism

Yoshihiro Kawahara,[†1] Kaoru Sezaki[†2] and Tohru Asami [†1]

Network management mechanism is one of the most essential elements for service providers to maintain the network services consistently. However, a single networked service sometimes consists of different service providers. As a result, it is difficult to share management information across the service providers and network domains. If a service fails in such a situation, the user of the service will not know why the service is unavailable now; He can't even detect the right person to ask for fixing it. In this paper, we introduce a management API based on REST architectural styles. This resource oriented management API allows software programs to access to management information across the domains over the Internet.

1. はじめに

コンピュータネットワークは，現代の社会を支える非常に重要なインフラストラクチャーである．昨今では，ユビキタスコンピューティングのコンセプトに見られるように，ネットワークを構成する機器，そしてネットワークへのアクセス手法が多様化し，網上を飛び交う情報も爆発的に増加している．ところが，現在のインターネットはこうした巨大な情報ネットワーク社会までを想定して設計されていなかったため，社会の変遷を担うことができずの可能性を伸ばすことができないでいる．20 年後，30 年後の情報ネットワーク社会を実現するためには，現実と仮想空間を融合しそれらを扱える「新世代」のネットワークそして，その管理・運用をしっかりと行う機構が必要不可欠である．

現在のインターネットで一般なネットワーク管理手法の原型は，1980 年代に当時の電話交換網をターゲットに研究開発された TMN (Telecommunication Management Network) にある．この技術的基盤は，ISO による OSI 管理プロトコルである CMIS (Common Management Information Service) / CMIP (Common Management Information Protocol) にある．しかし，実際のインターネットのネットワーク管理プロトコルとしては，当時の小規模なインターネットや企業ネットワークの現状に合わせて簡略化した SNMP (Simple Network Management Protocol)[1] が使われ，2 度の改良を経た後に今日に至っている．ASN.1[2] というプロトコル仕様言語の登場を背景に厳密に定義された SNMP は，ベンダー各社まちまちであった当時のカオス的なネットワーク管理の惨状を救ったという意味での貢献は大きい．しかしながら，文字通りの機能限定の管理プロトコルであったこと，セキュリティ面で管理ドメイン (範囲) の拡張に失敗したこと，アーキテクチャに柔軟性がなく時代に合った柔軟な拡張ができなかったことから，ネットワークの監視が主な役割となり，ルータ・スイッチの具体的制御は CLI (Command Line Interface) に頼って運用しているのが実情である．

†1 東京大学 大学院情報理工学系研究科
The University of Tokyo, Graduate School of Information Science and Technology
†2 東京大学 空間情報科学研究センター
The University of Tokyo, Center for Spatial Information Science

SNMP の限界は，標準化機関の IETF も認識しており，すべてのネットワーク機器のインタフェースを統一することを目指して NETCONF プロトコルの開発が進められてきた[3]．トランスポートプロトコルを SSL (Secure Socket Layer)，SSH (Secure Shell)，BEEP (Blocks Extensible Exchange Protocol) などに拡張し，SNMP の轍を踏むことを回避している．まだ実際の制御を決める管理オブジェクトの定義 (NETCONF では Content 層) の標準化は完成していないが，管理情報や設定情報に XML を使うアプローチと共に次世代プロトコルの方向性を示している．XML (Extensible Markup Language) は Web 系から発展したが，プロトコル仕様記述能力も高く，ASN.1 の後継技術としても普及しつつある．

一方，ネットワーク関連ビジネスは，1980 年代までの 1 社で全てのサービス提供を行う垂直統合型事業モデルから，同じ物理網上に，キャリア，ISP，ASP が重畳してサービス提供する水平分業モデルに移行し，同じアプリケーションサービスでも複数の事業者が介在するため，相互の運用情報の交換制御が難しく，円滑なサービス提供ができなくなってきた．この意味で，SNMP は垂直統合型ビジネス時代の遺物となりつつある．また，ネットワークの水平方向でも，1980 年代のコアネットワーク主体から，アクセスネットワーク，カスタマーネットワークへとインターネットの管理対象 (以下管理オブジェクトと呼ぶ) が広がっている．Ethernet Everywhere が代表的な現象である．アクセスネットワークも固定・無線双方で様々なシステムが提供されているが，カスタマーネットワークも従来の企業ネットワークだけでなく，ホームネットワークの管理まで踏み込まない限り，十分なエンド・ツー・エンドのサービス提供を行うことが難しくなってきている．また，ホームネットワークが組み込まれたことは，管理ドメインの拡大だけでなく，管理の対象となる機器の種類と量の爆発をもたらし，SNMP のように管理情報 (MIB) を標準化機関やベンダーが手堅く行うような手続きでは追いつかなくなっている．ユーザがユーザのニーズに合わせて使いたい通信対象の機能に合わせて自由に管理を定義し，それをネットワーク全体で再利用できるような枠組みが求められている．

この傾向は，今後のユビキタスネットワークの進展と共にますます強まると考えられ，端末やアプリケーションの量的拡大だけでなく，新しい機器への対応という質的拡大にも対応した，スケーラブルなネットワーク管理アーキテクチャが必要になっている．本稿では，これらの事情を鑑み，Web サービス連携技術である REST アーキテクチャスタイルに基づいたネットワーク管理 API を提案する．従来の手順指向のネットワーク管理ではなく，ネットワークのリソースを主体とした HTTP による管理情報の交換と制御が可能になり，ドメインをまたがったサービスの管理が容易になると期待できる．

2. 新世代ネットワークの管理機構に関する要求条件

提案するネットワーク管理機器機構を示す前に，我々が想定している管理機構への要求条件について述べる．

2.1 水平・垂直方向への多様さを持つこと

これからのネットワークは垂直・水平方向に様々な事業者が存在することが前提となる．したがって，同一のネットワーク管理コントロールプレーンにより，ユーザの身近な機器までを含め，物理層からアプリケーションまで柔軟に管理できることが必要となる．このことから，同一のルータ等の機器やサービスも見る人によって異なった制御や見え方（ビュー）を提供できなければならない．ASP から見えるルータネットワークと ISP のそれとでは，当然情報の粒度は異なる．

ASP から ISP のリソースをコントロールすることをある程度許すには，ビューとそれにアクセスできるクライアントの認証手続きが明らかになっていなければならない．柔軟に設定拡張でき，安心・安全なネットワークを構成できるビューの提供は，セキュリティ面での基盤的要求であるとともに，機器の導入が及ぼす水平・垂直方向の管理者への影響を最小限にするための機能でもある．これは，一般のユーザが直接ネットワークに働きかけて，そのとき，その場所，利用アプリケーションに合わせてネットワークサービスを構成する可能性への道を開く技術でもある．

2.2 トランスポートプロトコルへの非依存

IP ネットワークの管理，特にレイヤー 3 以下の管理で用いられてきた SNMP と比して，次世代ネットワーク管理プロトコルは，アプリケーションまでを含めた真のエンド・エンドの通信を管理できなければならない．

管理用のコントロールプレーンは，IP のような特定のトランスポートプロトコルを前提とすることができない，トランスポートプロトコル非依存のアーキテクチャでなければ，カスタマーネットワークまでの展開はおぼつかない．

センサネットワークのような，通信ノードが間欠動作するネットワークが出現したのも最近の動向である．管理したい瞬間に管理オブジェクトがネットワークにつながっている保証はなく，SNMP が前提した常時接続の暗黙仮定も崩れている．このように，ネットワークの進化にコントロールプレーンを追従させるには，トランスポートプロトコル非依存は必須の条件である．

2.3 スケーラビリティと拡張性

通信プロトコルやアプリケーションは，日進月歩で多様化する．このため，管理オブジェクトの定義は，中央集権的に行うのではなく，分散的に行わざるを得

ない．スケーラブルなネットワーク管理を実現するには，この種のローカルな機器やアプリケーションの拡張をシステム全体でいかに収容し利用していくかという問題を解決できなければならない．また，前述したようにネットワークの管理の対象は，IP 通信機器とは限らない．基本的にネットワーク上に物理的にあるいは仮想的に存在するすべてのオブジェクトが管理対象と覚悟しないと，今後のネットワークの進化に対応することはできない．この意味で，管理用のネットワーク（コントロールプレーン）に関して，カスタマーネットワークのような通信リソースの限られているネットワークに対しても安定に管理できるような構成を持つことが要求される．ディペンダブルなネットワークを管理するネットワーク管理アーキテクチャはディペンダブルなものでなければならない．

また，ネットワーク管理技術では，既存ネットワークやその管理システムの存在を踏まえた移行技術がない限り，いかなる新技術も導入することはできない．既存管理システムと相互接続性が前提とされる中で，革新的な技術を導入しなければならないところが，この分野の特殊性であり課題のひとつでもある．

3. REST アーキテクチャスタイルによる管理

本論文では，2 節の要求条件を満たすネットワーク管理コントロールプレーンを構成するためのキーテクノロジとして，Web サービスの連携技術である REST アーキテクチャスタイルによる API に着目する．

3.1 REST アーキテクチャスタイルと Web サービス連携

REST アーキテクチャスタイルとは，Web サービスの連携技術として注目を集めているリソース連携手法である[4]．REST アーキテクチャスタイルが最も特徴的な点は，ネットワーク上の機能を「手順」ではなく，「リソース」を中心に考えて連携させる点である．リソースには一意な URI が割り当てられ，HTTP を利用して通信する．HTTP は WWW で利用されている上位層プロトコルであるが，昨今ではサーバから PC，携帯電話組み込み機器まであらゆる計算機で利用されており，またネットワーク管理の観点からドメインを超えた通信が比較的緩く実現できる唯一の上位層プロトコルでもある．

この REST API を用いた Web サービスマッシュアップの具体的な事例としては，ある検索結果に基づいてレストラン情報と地図情報を同時に表示したり，YouTube など動画コンテンツと同じキーワードの商品を amazon.com などより取得して表示するなど，単体の Web サービスでは提供できなかったユーザにとって高い利便性を与える複合サービスがあげられる．これらのマッシュアップされたサービスの多くは，大手企業によってトップダウン式に提供されるのではなく，異なる事業者が提供するサービスを一部の個人的な開発者がアドホックかつ，水平分散的に連携することにより，サービスが構成されている．

REST(REpresentational State Transfer) とは分散システムにおいてコンポーネントのスケーラビリティと独立性を最大化しつつ通信量と遅延を最小化するようなアーキテクチャスタイルのことを指す．REST は，初めはアーキテクチャの原則と制約の集まり (後述) を指していたが，次第に XML や HTTP を使った簡易なウェブベースのインタフェースのうち Web サービスの SOAP プロトコルのような MEP（Message Exchange Pattern; SOAP ノード相互のメッセージ交換のパターンを確立するための雛型）ベースの特別な抽象化をしないもののことを，大まかに意味する用語として使われるようになった．REST においては名前を付けることができるあらゆる情報を “リソース” と呼び (例:東京地方の今日の天気情報)，リソースの実体 (晴/雨という情報）は時間や条件によって変化するかもしれないが，本質的に示す意味は普遍で一定であると考える．すべてのリソースには URI などの識別子が与えられ，識別子に対して POST/PUT/DELETE/GET の 4 つのアクションを許すことでリソースの作成，変更（更新），削除，状態取得を行う．

この REST の考え方を新世代ネットワークのネットワーク管理に適用すると，センサ情報や，コンテキスト，表示装置やロボットの状態といった，従来 MIB で表現されてきた情報だけでなく，それらを含む個々のネットワーク自体をも独立したリソースと考え，管理サービスはそのリソース同士の連携として見なすことができる．REST の最大のポイントはリソースのセマンティクスをアプリケーションにゆだね，ヘテロジニアスなリソースが一意に識別できる相互接続する点のみを規定することにある．これにより，サービスの呼び出し粒度を大きく保ちネットワークでの状態管理も不要になるため，SNMP が抱えてしまったような画一性を排除できる．さらに RPC や DCOM，CORBA などの従来の分散オブジェクト技術ともことなり非常にスケーラビリティの高い構成が可能になる．

3.2 相互接続点としての Active Proxy

従来のアーキテクチャでは，管理ドメイン間の連携が非常に実現しにくく，このため，サービスをエンド・エンドで管理することは特定の通信事業者のネットワーク内に限られ，インターネットの規模では実現できなかった．本提案手法は，基本的に相互接続点の規定の集合であるため，管理ドメインはあるポリシーで接続が規定されている相互接続点の部分集合とみなすことができる．管理ドメイン間の連携は，ドメイン間の相互接続ポリシーを決めれば導出可能である．このようなヘテロジニアスな管理ドメインの存在をアーキテクチャに最初から内在させている点で，SNMP 等の一様な管理アーキテクチャと最も異なる点である．この

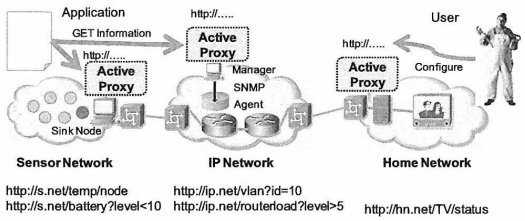

図 1 Active Proxy による管理と制御

結果，インタードメインの管理で一番の課題であった，同じリソースを相手のリソースによって見え方を変えるような機能 (ビュー) も容易に実現可能である．本研究では，このようなポリシー制御を容易にするため，REST を拡張して，Active Proxy サーバの概念を追加し，クライアントへのビューの制御をより明示的に行うことができるようにする (図 1)．これにより，たとえばある管理リソースについて個別のユーザによって管理操作をすべて許すが，別のユーザには一部の操作のみに限定するとか，そもそも存在を察知されないようにするといったような柔軟な管理が可能になる．

REST アーキテクチャスタイルではリソース間の連携はミクロには，クライアントサーバ方式を踏襲しているため，階層化や負荷分散，レガシーシステムのラッピング，プロクシサーバの利用が可能である．したがって，新世代ネットワークを構成する各ネットワークで既にデファクトスタンダードとなっている，識別子に対して POST/PUT/DELETE/GET の 4 つのアクションを許す構成は，MIB に対する SET/GET/GETNEXT の 3 つのアクションと TRAP を基本とする SNMP とも親和性が高く，また CLI による管理インタフェースとも容易に共存可能である．

SNMP の MIB と異なり各機器の URI は FQDN がユニークであれば，誰でも柔軟なアサインメントが可能である．そして (3) の性質により管理者と管理対象機器が HTTP を用いることで管理対象オブジェクトに確実につながることが保障される．そして，(1) の XML の性質により，性質の異なる管理対象機器が登場した場合には，新たな XML スキーマを定義し，追加することが可能であるほか，レガシーシステムをラッピングして提供することも容易である．こうした機構を用いることにより，管理オブジェクトの定義を，分散的に行うことができるようになる．スケーラブルなネットワーク管理を実現するには，ローカルな機器やアプリケーションの拡張をシステム全体でいかに収容し利用していくかという問題を解決できなければならない．

本方式は，ネットワークの管理の対象を IP 通信機器に限らず，ネットワーク上に物理的にあるいは仮想的に存在するすべてのオブジェクトを管理対象とし，日進月歩で多様化する通信プロトコルやアプリケーションを統一的に管理することが可能である．

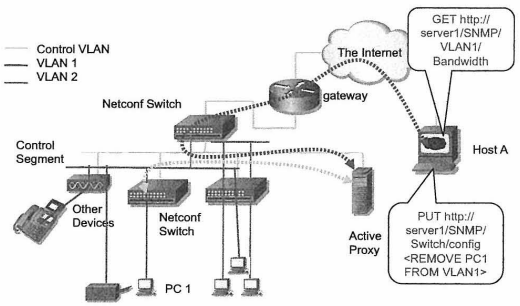

図 2 SNMP, NETCONF に対応した Active Proxy

4. Active Proxy の設計と動作

本節では，Active Proxy の適用例として，SNMP による機器のモニタリング，NETCONF による機器設定操作，センサネットワークの設定管理，の 3 つの機能を実装するために必要な設計について示す．

4.1 機器構成

図 2 は，SNMP による機器モニタリングと NETCONF による機器設定操作が可能な Active Proxy が設置されたローカルエリアネットワーク (LAN) の例である．まず，前提として，この LAN は SNMP および NETCONF に対応した Swith から構成されている．そして，制御管理用の “Control VLAN” と業務用の “VLAN 1” および “VLAN 2” が利用されている．Control VLAN には，Active Proxy サーバが接続されている．また Active Proxy は Global な HTTP プロトコルによりインターネットからアクセス可能であり，その URI は http://server1 であるとする．

Host A はこの Active Proxy を介して LAN の内容を管理したいホストである．Host A はインターネット経由で Active Proxy にアクセスする．要求を受けた Active Proxy は適当な手段[*1]で Host A を認証した後，管理要求内容を解析し，SNMP または NETCONF プロトコルにより LAN 内のネットワーク機器にアクセスし応答を得る．応答が Active Proxy に返されると，Active Proxy はその内容を XML フォーマットに加工し，Host A に返す．

同様に，センサネットワークの管理を行う際，Active Proxy はセンサネットワークの Sink ノードとインターネットに接続する形で設置する．こうすることでインターネット上のホストからの管理要求をセンサネットに対して発行することができる．

4.2 SNMP 用 URI シンタックス

現在の設計では，SNMP の Get,Get-next,Set コマンドを実装し，snmpget, snmpset, snmpwalk, snmp-

*1 SSL によるサーバ・クライアント認証，WSSE 認証等，様々な認証および認可機構が考えられる．

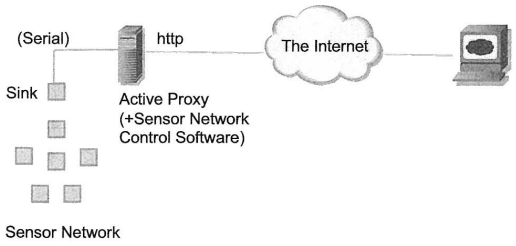

図 3 センサネットワークに対応した Active Proxy

bulkget コマンドと同等の機能を提供する*1．SNMP 情報にアクセうするための Active Proxy リソースの URI シンタックスは共通して以下の書式をとる．

URI Syntax

{GET,PUT,POST,DELETE}
http://*host*/tbsnmp/*operation*/*agent*/*community*
/*parameters*...

ここで，*host* は Active Proxy の FQDN，*operation* は目的のコマンド，*agent* は SNMP の管理対象機器の識別子，*community* は SNMP コミュニティ名 *parameters* はコマンド固有のパラメータである．これら URI に対して HTTP の GET/POST/PUT/DELETE のいずれかのアクションを行う．

まず，SNMP Get は機器の情報を取得するコマンドである．

SNMP Get

GET http://host/tbsnmp/get/agent/community
/{oid,mibtree}
例)
GET http://host/tbsnmp/get/192.168.10.2/public
/system.sysContact.0
GET http://host/tbsnmp/get/localhost/private
/.1.3.6.1.2.1.1.6.0

同様に Set コマンドを用いて値を設定することも可能である．

SNMP Set

{POST,PUT}
http://host/tbsnmp/set/agent/community
/{oid,mibtree}

*1 SNMP の情報取得ではない，通知機能である Trap は Weblog Ping 機能を使うことで実現する．

getnext, getbulk

GET http://host/tbsnmp/getnext/agent/community
/{oid,mibtree}
GET http://host/tbsnmp/getbulk/agent/community
/{oid,mibtree}?paramname=param

getnetxt は探している OID の次のものを返し，これを利用した getbulk では繰り返し方法を取得する．

4.3 NETCONF 用 URI シンタックス

NETCONF は IETF で標準化作業が行われている，XML(eXtensible Markup Language) で記述された管理情報によるネットワーク管理プロトコルであるが，現在いくつかの実装が利用可能である．今回は Alaxala 社が提供する AX-Open Networking - Application Programming Interface (AX-ON-API) を利用した場合の実装について示す．

共通する Syntax は以下の通りである．

URI Syntax

{GET,PUT,POST,DELETE}
http://*host*/tbnetconf/*target equipment*
/*operation*...

Operation には，Node(ネットワーク機器) 操作，Line(回線) 設定，VLAN 設定，Link Aggregation, Abstraction Port, Accesslist, 経路制御，MAC アドレス情報取得等が可能であるが，本論文では一例として VLAN 設定の operation に限って示す．

VLAN 新規作成例

POST http://host/tbnetconf/*target equipment*
/vlan/*vlanid*

この場合 Vlan ID が *vlanid* である VLAN を作成する．作成する VLAN には Port VLAN, Tag VLAN, Protocol VLAN, MAC VLAN, Subnet VLAN など複数の種類があるが，詳しい設定内容は XML で記述し URI に POST する．

また VLAN の設定内容に関して変更を行う際には POST ではく PUT を利用する．VLAN 自体を削除したいときは URI に対して DELETE し，単に設定内容だけを確認したいときは GET を利用する．

4.4 センサネットワークの管理

センサネットの管理には，SNMP 等の既存のネットワーク管理手法と異なる機能が求められる．第一にセンサネットは不安定なネットワークであるため，稼動状況を管理するには，各ノードのバッテリの残量や経路情報を知る必要がある．また，センサネットのノード数は多数になるので，ノード単体ではなく，グループとして管理できることも必要である．グループの作成には，ノードの設置場所を基準にした Location グループや，センサの型に応じた Type グループなど，

複数のグループへの対応が必要である．さらに，運用開始後にセンサネットを利用するアプリケーションを開発する開発者がノードを検索し，独自のグループをあとから追加定義できることも必要である．

センサネット管理 URI 例

URI1	http://host/
URI2	http://host/sensornetA/
URI3	http://host/sensornetA/battery/
URI4	http://host/sensornetB/battery/?id=3&id=5
URI5	http://host/sensornetA/topology/
URI6	http://host/groups/
URI7	http://host/groups/location/livingroom
URI8	http://host/groups/my-group
URI9	http://host/groups/location/my-desk

まず，トップレベルの URI1 に GET することで Active Proxy により管理されているプロパティリストを XML で取得できる．この場合，sensornetA，sensornetB 及び groups へのリンクのリストが得られる．URI2 は，sensornetA のプロパティリストを指す．その中に URI3 が定義されており，この URI が sensornetA の各ノードの電源状況を指す．また，URI4 を GET メソッドで呼べば，特定のノード ID の電源の状況を絞り込んで取得できる．URI5 は，sensornetA のトポロジーを指す URI である．トポロジーは，各ノードとその親ノードの ID を対応付けた表により表される．

URI6 以降はノードをグループとして扱うための URI である．URI6 は Active Proxy に存在しているグループ全てのプロパティリストを指す．この URI に GET メソッドでアクセスすると，グループのリストとリンクが取得できる．URI7 は location グループの中の livingroom グループのプロパティリストを指す．このように，グループのグループも許す．URI8，URI9 は，自分で設定したグループへの URI である．URI8 のようなグループを設定するには，URI6 に，そのグループに入れるノード ID のリストを記述し，POST する．また，設定ずみのグループに変更を加えるときは PUT メソッドで，削除するときは DELETE メソッドでアクセスする．URI9 も URI8 と同様であるが，URI9 は location グループの中に my-desk というグループを定義している．

5. おわりに

今後，コンピュータネットワークは，端末やアプリケーションの数が増加するだけでなく，センサネットワークやホームネットワークと言った新しい機器，新しいドメインへの質的拡大が期待されている．こうしたヘテロジニアスなネットワークをまたいだ連携型のサービスが展開されるようになると，その運用管理には，これまでの前提と全く異なるスケーラブルなネットワーク管理アーキテクチャが必要になる．本稿では，これらの事情を鑑み，Web サービス連携技術である REST アーキテクチャスタイルに基づいたネットワーク管理 API を提案した．

従来のアーキテクチャでは，管理ドメイン間の連携が非常に実現しにくく，このため，サービスをエンド・エンドで管理することは特定の通信事業者のネットワーク内に限られ，インターネットの規模では実現できなかった．本提案手法は，Active Proxy による相互接続点の規定の集合であるため，管理ドメインはあるポリシーで接続が規定されている相互接続点の部分集合とみなすことができる．管理ドメイン間の連携は，ドメイン間の相互接続ポリシーを決めれば導出可能である．このようなヘテロジニアスな管理ドメインの存在をアーキテクチャに最初から内在させている点で，SNMP 等の一様な管理アーキテクチャと最も異なる点である．

また，センサネットワークの構成例において示したように，管理を行うユーザが管理対象機器を絞り込み，絞り込んだ結果に “新たな管理リソース” として別名の URI を定義することも可能である．すなわち，インタードメインの管理で一番の課題であった，同じリソースを相手のリソースによって見え方を変えるような機能 (ビュー) も容易に実現可能である．

現在は，SNMP,NETCONF, ホームネットワークやセンサネットワークを含めた異種ネットワークを統一的に制御管理するための Active Proxy 機能を実装中である．

参考文献

1) Best Current Practice - Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework, RFC 3584
2) Information processing systems - Open Systems Interconnection, “Specification of Abstract Syntax Notation One (ASN.1)”, International Organization for Standardization, International Standard 8824, December 1987.
3) IETF RFC4741：NETCONF Configuration Protocol (2006.12)
4) Leonard Richardson, Sam Ruby: RESTful Web Services, ISBN 10: 0-596-52926-0, O’Reilly Media, Inc., May 2007

謝辞

本研究は，(独) 情報通信研究機構 H19 年度委託研究「新世代ネットワークの構成に関する設計・評価手法の研究開発」の一環として実施された．