

形式の異なる IP ルータの構成情報を相互形式変換する方式の提案と評価

堀賢治[†] 吉原貴仁[†] 井戸上 彰[†][†]株式会社 KDDI 研究所

1 はじめに

企業ネットワークにおけるローエンドの IP ルータに求められる機能、例えば、経路制御プロトコルや VLAN は標準的であり、類似機能を備える多様な機種が市場に存在する。IP ルータの設置や置換、管理を行う管理者は、費用に対して、パケット転送などの性能や安全性の高い機種を採用する。このため必然的に異機種混在環境となる。このような環境で、管理者は形式の異なる多様な IP ルータの構成情報を取り扱う必要がある。各形式への習熟や、異機種への置換に伴う構成情報の形式変換作業など、負担が大きい。このような負担を緩和するため、従来、機種に依存しない統一的な構成情報の形式(以下、統一形式と呼ぶ)を導入し、統一形式から機種固有形式へと自動変換する方式がある。しかし管理者が新たに統一形式に習熟することを前提としており、異機種への置換に伴う構成情報の形式変換作業などに必要な、機種固有形式から他の機種固有形式への変換を行うことは想定されていない問題がある。そこで本稿では、機種固有形式間の変換も可能な構成情報の相互形式変換方式を新たに提案する。また、提案方式を評価するためのプログラムを実装し、構成情報の変換成功度合いの観点から評価する。

2 従来方式とその問題点

NETMOD[1]のように構成情報の形式を統一化・標準化し、将来的にあらゆる IP ルータを統一形式の構成情報に対応させる取り組みがある。また、PRESTO[2]のように、統一形式の構成情報や、ネットワーク全体に望まれる振舞い(ポリシー)の記述から、機種固有の構成情報を自動生成する取り組みもある。これにより、管理者は統一形式の構成情報にのみ習熟することで、管理用コンピュータが機種固有形式の構成情報を自動生成するため、負担を緩和できる。

しかし[1]は、1章に先述の既存の異機種混在環境では必ずしも有効でない。また[2]は、統一形式の構成情報から機種固有の構成情報を生成する一方向の変換のみ想定している。機種固有の形式から、統一形式や他の機種固有形式へと変換することは想定されていない。このため管理者は新たに統一形式に習熟する必要がある。また、異機種への置換に伴う構成情報の形式変換作業や、過去に作成した構成情報の流用も困難である。そこで以下、機種固有形式の構成情報を、他の機種固有形式との間で相互変換する構成情報変換方式を提案する。

3 提案方式

3.1 基本方針

機種固有形式から他の機種固有形式へと変換する際、一度、統一形式に変換し、これを他の機種固有形式に再度変換する方式(以下、統一形式仲介方式と呼ぶ)と、機種固有形式間で直接相互変換を行う方式(以下、直接相互変換方式と呼ぶ)が考えられる。一方、1章に述べたように、企業ネットワークにおける IP ルータの機種は多様である。このため、本方式による構成情報形式変換処理を実装された「変換プログラム」が、対応機種数(以下、 N とする)の増加に対しスケーラビリティの高い構造、具体的には変換プログラムの総コーディング量の増加が少ない方式が望ましい。ここで、ある 2 形式(直接相互変換方式であれば機種固有形式同士、統一形式仲介方式であれば機種固有形式と統一形式)間の変換処理は、各一つの変換プログラムが行うモジュール型の構造を仮定すると、変換プログラム数 N に対する増加が少、かつ、各変換プログラムのコーディング量が少ない方式が望ましい。まず変換プログラム数は、統一形式仲介方式では $O(N)$ 、直接相互

表 1 機種固有形式の構成情報例(Cisco 社製 1812J 型用)および統一形式への変換例

機種固有形式	interface FastEthernet0.10 encapsulation dot1Q 10 ip vrf forwarding sales ip address 192.168.1.1 255.255.255.0 !
統一形式	<interface> <PhysicalPort> <PhysicalPortNo>0</PhysicalPortNo> <VlanNo>10</VlanNo> <encapsulationDot1Q>10</encapsulationDot1Q> <IpVrf> <Forwarding>sales</Forwarding> </IpVrf> <IPAddress-static> <address>192.168.1.1</address> <subnetmask>255.255.255.0</subnetmask> </IPAddress-static> </PhysicalPort> </interface>

変換方式では $O(N^2)$ となる。

次に各変換プログラムのコーディング量は、変換プログラムが解析可能な IP アドレスや経路情報等の設定命令(以下、設定命令と呼ぶ)、例えば、表 1 の「機種固有形式」の例では interface に相当する設定命令の数に比例すると考えられる。1章に先述の環境では機種間の機能差が小さいことから、対応する設定命令数に大差は無い。よって、変換プログラムのコーディング量は機種によらずほぼ一定とみなすことができる。

以上から、本稿ではスケーラビリティの高い、統一形式仲介方式を基本とした方式とする。

3.2 XML による統一形式

統一形式として NETMOD[1]に従う Extensible Markup Language (XML)形式を用いる。これにより XML 文書構造の定義(XML スキーマ等)を流用できる。しかし、[1]やこれに関連する IETF ドラフトによる構成情報モデルの定義は途上であり、流用可能な定義のない設定命令も多い。これらに関する XML タグは独自に定義する。表 1 に統一形式の例を示す。

3.3 変換動作

機種固有形式から統一形式への変換動作を以下に示す。逆変換(統一形式から機種固有形式)の場合も、解析・置換する対象が XML タグとなるのみで、原則同様な処理となる。

(1) 機種固有形式の構成情報を記述したテキストファイル名とその機種を、変換プログラムを呼び出す補助プログラムに与える。機種は下記(2)で必要となる。

(2) 上記(1)で指定した機種に対応する変換プログラム部が呼び出され、入力された機種固有形式の構成情報から統一形式への変換処理を行う。変換結果は上記(1)とは別のテキストファイルに保存される。

ここで(2)の変換処理は、(i)構成情報を解析し設定命令を探索する処理、および(ii)設定命令の置換処理とからなる。

(i)は、ISC bind 等多くの既存プログラムと同様に、構成情報の設定命令とその終端記号を探索し、発見した設定命令ごとにその属性名を探索するトップダウン型の処理を行う。表 1 の「機種固有形式」の例では、設定命令として interface が、その属性名として FastEthernet, encapsulation, ip vrf 等が、さらに終端記号として“!”(エクスクラメーション)が該当する。

尚、例えば、古河電工社製 F80 型ルータでは line 設定命令から interface 設定命令を参照することで VLAN が定義されるが、このように、他の設定命令の参照により定義される設定命令には対応していない。

(ii)は、(i)で発見した設定命令と属性名を、統一形式のXMLタグとそのタグの値にそれぞれ置換する。例えば表1の例では、interface 設定命令と終端記号“!”が<interface></interface>というXMLタグに変換されている。

表1の<PhysicalPort>や<subnetmask>のように、機種固有形式に自明には現れないXMLタグもある。これらは統一形式から機種固有形式への置換処理を容易にする狙いで挿入される。<PhysicalPort>は、特定の設定命令(interface)と属性値(FastEthernet)の組み合わせを検出した場合に挿入される。<subnetmask>は、特定の属性値(ip address)を検出した場合、後続する属性値、この場合はサブネットマスクを、IPアドレスと対して格納するために挿入される。

尚、変換プログラムが対応しない設定命令は統一形式に変換されない。また、ある機種固有形式から統一形式を経て他機種固有形式へと変換する際、変換前後の機種固有形式で属性値の種類または数が異なる場合(e.g., 1812J型ではinterface 設定命令に属性値として bandwidth が指定可能だが、F80型では指定できない)がある。この場合、統一形式から機種固有形式への変換時に、操作者が認知し任意に修正できるように、変換前の構成情報の当該箇所を別のテキストファイルに出力する。また変換プログラムが対応する機種同士であっても、意味的に共通の設定命令や属性名、属性値が提供されない場合も同様となる。

4 実装

提案方式の有効性を評価するため、評価用プログラムを実装した。評価用プログラムは(i) 構成情報を記述したファイルの入出力や保存等を行うための簡易な Graphical User Interface(GUI)を提供する、IP ルータ機種に非依存な部分と、(ii) 3章に述べた IP ルータ機種依存の「変換プログラム」(Java クラスモジュール)とからなり、Java 言語(J2SE 1.4.2_16, Sun Microsystems Inc.)で実装している。

機種はCisco社製1812J型、古河電工社製F80型、およびAlaxala社製AX2430S型に対応する。尚、その他多くの機種はCisco社に類似した設定命令体系を備えるため、Cisco社の構成情報形式への対応を優先し、変換プログラム開発が容易となるよう実装した。対応機種は今後拡大予定である。

設定命令は、1812J型における interface, access-list, ip route, router, crypto 等に相当する、利用頻度の高い11種類に対応する。設定命令の属性名、属性値、および属性値の型は機種毎に異なる。例えば、1812J型の interface 設定命令の第1番目の属性値としてはFastEthernet, Dialer, Loopback, BRIなど20種類すべてに対応する。

表1に示した統一形式は、1812J型における interface 設定命令を含む構成情報を、統一形式へと変換した例である。

5 評価

5.1 評価方法

評価用プログラムをWindows XP sp3 OS(Microsoft Inc.)を搭載するPC(Thinkpad T60p, Lenovo Inc.)で実行し、構成情報変換の成功度合いを以下のように評価した。

異なる2つの機種固有形式間の変換前後で構成情報の意味内容が保存されているほど成功と言える。ここで異機種の構成情報の比較検証は手作業にならざるを得ず主観的となる。よって機械的な比較検証による差異評価手段が必要である。そこで一度、構成情報を下述の「中間形式」に変換した後、再度、元の機種の形式に変換することで、変換前後の意味内容の差異を、両形式の差異箇所として機械的に比較した。

(1)各機種(1812J型, F80型, AX2430S型)固有形式の構成情報(以下、変換元構成情報と呼ぶ)を統一形式へと変換する。この変換元構成情報の形式を変換元形式と呼ぶ。

対象とする変換元構成情報は、表1の機種固有形式の例のように、設定命令を一つと、その属性値を含むような合計3種類である。具体的には、1812J型における interface

表2 変換成功度合い(差異行数)の評価結果

	変換元形式	中間形式	差異の発生した構成情報および差異行数[行]
(a)	1812J	1812J	差異なし
(b)	1812J	F80	interface FastEthernet (1[行]) interface vlan (4[行])
(c)	1812J	AX	interface FastEthernet (3[行]) interface vlan (3[行])
(d)	F80	1812J	差異なし
(e)	F80	F80	差異なし
(f)	F80	AX2430S	interface FastEthernet (3[行]) interface vlan (5[行])
(g)	AX2430S	1812J	差異なし
(h)	AX2430S	F80	差異なし
(i)	AX2430S	AX2430S	差異なし

FastEthernet, hostname, interface vlan に相当する設定命令および属性とする。これらは利用頻度が高く、対応3機種とも共通または類似の機能を備える。

(2)上記(1)で得られた統一形式を再び各機種固有形式の構成情報(以下、中間構成情報と呼ぶ)に変換する。ここで、対応する3形式すべてについて、それぞれ中間構成情報の形式(以下、中間形式と呼ぶ)として利用する。具体的な変換元形式と中間形式との組み合わせは表2の各行に示される。

(3)上記(2)で得られた中間形式の構成情報を、再度、統一形式へと変換する。

(4)上記(3)で得られた統一形式(合計9個)を変換元形式へと変換する。以下、この変換で得られた構成情報を最終構成情報と呼ぶ。すなわち、最終構成情報と変換元構成情報の形式は同一である。

(5)GNU diffにより上記(1)の変換元構成情報と上記(4)の最終構成情報との差異行数を求め、その少なさを変換成功度合いの指標とする。

5.2 評価結果と考察

表2に差異行数の評価結果を示す。変換元形式と中間形式が同一の場合(表2(a),(e)および(i))、差異は生じていない。それ以外の6つの場合(表2(b),(c),(d),(f),(g)および(h))のうち、3つの場合(表2(b),(c)および(f))で差異を生じている。

後者の場合について中間形式での構成情報も確認した結果、変換前後の構成情報形式の間で、設定命令が一对一対応でも、属性値の種類・数が異なる場合(interface FastEthernet)に差異が生じていた。

一つの設定命令が複数の設定命令へと変換される場合(interface vlan)、中間形式から統一形式への変換(5.1節(3))において、変換元形式からの構成情報意味内容の差異が発生し、差異行数を増やす原因となっていた。これは構文解析処理が他の設定命令への参照に対応していない(3.3節)ためである。他の設定命令を参照する可能性がある設定命令を発見した場合に、その参照先を探索し紐付け情報を保存しておくといった対策をとることで、差異行数削減の可能性はある。

6 おわりに

本稿では構成情報の双方向変換方式を提案し、評価用実装を構築して変換成功度合いの評価結果を示した。今後は対応機種を拡張すると共に、他の設定命令を参照する設定命令への対応を図る。最後に、日頃ご指導頂く(株)KDDI研究所秋葉所長、ならびに鈴木執行役員に感謝する。

参考文献

- [1] D. Partain, et al., “NETCONF Data Modeling Language (netmod) Charter,” <http://www.ietf.org/html.charters/netmod-charter.html>, August 2008.
- [2] W. Enck, et al., “Configuration Management at Massive Scale: System Design and Experience,” Proceedings of 2007 USENIX Annual Technical Conference, pp.73-86, April 2007.