

## SNMP/OSI管理ゲートウェイにおける効率的な管理操作変換の実現

1 F - 4 黒木哲也 堀内浩規 杉山敬三 小花貞夫 鈴木健二  
国際電信電話株式会社 研究所

## 1. はじめに

SNMPのマネージャを用いて、CMIPの管理インターフェースを具備するTMN（電気通信管理網）の伝送装置等の監視／制御を可能とするため、ワークステーション上で管理操作および管理情報の変換を行うSNMP/OSI管理ゲートウェイを実装している<sup>[1]</sup>。本稿では、マネージャ、ゲートウェイ、エージェント間の管理操作回数を削減する、効率的な管理操作変換の実現について報告を行う。

## 2. SNMP/OSI管理ゲートウェイにおける管理操作変換の概要

SNMP/OSI管理ゲートウェイでは、SNMPのオブジェクトとTMNのOSI管理の管理オブジェクト(MO)クラスとの対応づけを記述したSNMP/OSI管理情報定義対応テーブルを持ち、管理操作の変換は、このテーブルに基づいて行う。ここでは、OSI管理情報定義(GDMO)のMOクラス定義とSNMPのオブジェクト定義の対応づけに、NMF(Network Management Forum)のIIMCの規則を用いている<sup>[2]</sup>。マネージャによるSNMPのオブジェクトへのアクセスは、TMNエージェントの属性値へのアクセスに変換される。例えば、マネージャがオブジェクトの値の取得を要求した場合、ゲートウェイでは、GetRequestのパラメータに指定されるオブジェクトの識別名(OID:Object Identifier)から、対応づけられるエージェントのMOクラス、属性ID等を求め、これらのパラメータを指定したM-GETreqを発行する。

## 3. 管理操作変換での課題

[課題1] ゲートウェイ／エージェント間のアクセス回数の削減

GDMOの構造形の属性型は、複数のオブジェクトに対応づけるため、マネージャが属性値を検索する際には、多数の管理操作を発行する可能性があり、ゲートウェイからエージェントへのアクセスも増加する。また、内容の変更が発生しない属性値の場合には、マネージャからの検索要求毎にゲートウェイがエージェントへアクセスするのは冗長である。従って、属性値の検索時にゲートウェイとエージェント間の管理操作の数を削減する必要がある。

[課題2] マネージャ／ゲートウェイ間のアクセス回数の削減

属性型が構造形の場合、そのメンバの集合を表すサイドテーブルと、サイドテーブルとの接続を示すポインタおよびオブジェクトに対応づける。また、メンバが構造形のネストを繰り返す場合、多段のサイドテーブルとその接続を示すポインタに対応づけ

*Efficient Mechanism for Management Operation Conversion in SNMP/OSI Management Gateway*

Tetsuya KUROKI, Hiroki HORIUCHI, Keizo SUGIYAMA,  
Sadao OBANA and Kenji SUZUKI KDD R&D Laboratories

る。このため、サイドテーブルのオブジェクトにアクセスするには、先頭から順にポインタをトレースする必要があり、アクセス回数が増加するので、これを削減する必要がある。

以下にこれらの課題の解決方法を述べる。

## 4. 効率的な管理操作変換の実現手法

マネージャ、ゲートウェイ、エージェント間でのアクセス回数を減少させ、効率的な管理操作変換を行うため、以下のキャッシュ、レプリカ(複製)、リンクオブジェクトを導入する。

(1) キャッシュ (課題1への対処)

マネージャが構造形の属性値の検索を行う場合、連続的にGetRequest/Get-NextRequestを発行する必要がある。このため、GetRequest/Get-NextRequestに対応づけられたM-GETreqによって得られた属性値が構造形の場合、値をキャッシュとしてゲートウェイに保持する。次からの検索要求が、キャッシュの属性値に対するもの場合、エージェントへはアクセスせずに、キャッシュの情報を返送する(図1)。

キャッシュは、マネージャ毎に一個の構造形の属性値を保持し、キャッシュ以外の属性値にアクセスが発生した場合や、SetRequestが発生した場合には、キャッシュを消去する。

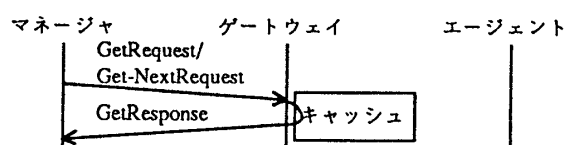


図1 キャッシュによる効率化

(2) レプリカ (課題1と2への対処)

マネージャが、構造形の属性値の一部のメンバに対応するSNMPオブジェクトの値を変更するには、SetRequestの発行にあたって事前にGetRequest等で、同一属性値内の他のメンバに対応する全てのオブジェクトの値を取得する必要がある。これによりマネージャ、ゲートウェイ、エージェントのアクセス回数が増加する。そこで、GetRequest/Get-NextRequestの実行で取得した属性値をレプリカとして保持し、同一属性値に関するSetRequestが発生した場合、指定されたオブジェクト(構造形のメンバ)以外の値をレプリカから補うことで、これを解決する(図2)。また、マネージャが構造形の属性値に関するGetRequest/Get-NextRequestを要求し、キャッシュが存在しない場合には、レプリカを利用することでエージェントへのアクセスは行わない。

レプリカは、エージェントの属性値毎に作成の要／否を指定可能とする。また、属性値の異なる複数個のレプリカを管理可能で、規定する最大の数を越えた場合、最も古いレプリカを削除する。

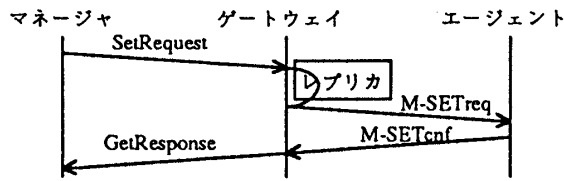


図2 レプリカによる効率化

(3) リンクオブジェクト (課題2への対処)

サイドテーブルで使用しているインデックス値を、その階層構造に従って保持するリンクオブジェクトを新たに定義する。マネージャがリンクオブジェクトの値を取得することで、サイドテーブルの接続関係がわかり、ポインタをトレースせずに、直接SNMPのオブジェクトにアクセス可能となる。リンクオブジェクトのASN.1定義を図3に示す。

```
LinkObject ::= SEQUENCE OF SEQUENCE {
    index INTEGER
    child SEQUENCE OF LinkObject OPTIONAL }
    * index はインデックス値を、child はサイドテーブルの階層を示し、最終段に至ったとき OPTIONAL を適用する。
```

図3 リンクオブジェクトのASN.1定義

リンクオブジェクトは、GDMOのMOクラス毎に定義し、リンクオブジェクトの値を取得する場合には、構造形の属性値に対応するオブジェクトのOIDと、その属性を含むMOクラスに対応するリンクオブジェクトのOIDをペアで指定する。

上記(1)、(2)、(3)の手法を用いたGetRequest/Get-NextRequestとSetRequestの処理の流れを、それぞれ図4および図5に示す。

5. 考察

キャッシュ、レプリカとリンクオブジェクトによる管理操作のアクセス回数削減の効果は、ITU-T勧告X.721の属性型 weekMask 使用時(サイドテーブル3段、インスタンス数1個)では、以下ようになる。

(1) キャッシュ: メンバ hour の値取得時には、キャッシュ不使用では4回のM-GETreq/cnfが必要であるが、使用時には1回のM-GETreq/cnfで検索可能となる。

(2) レプリカ: メンバ daysOfWeek の値設定時には、レプリカ不使用では3回のGetRequestとそれぞれ1回のM-GETreq, M-SETreqの発行が必要だが、使用時は1回のM-SETreq発行のみとなる。

(3) リンクオブジェクト: メンバ minute の値取得時には、リンクオブジェクト不使用では4回のGetRequestが必要であるが、使用時には2回のGetRequestで検索可能となる。

構造形のネストの繰り返し数やインスタンス数が多くなるに従って、さらにその効果は増大する。

6. おわりに

本稿では、SNMP/OSI管理ゲートウェイの管理操作変換において、キャッシュ、レプリカ、リンクオブジェクトを導入することにより、マネージャおよびエージェントとの間のアクセス回数を削減して、

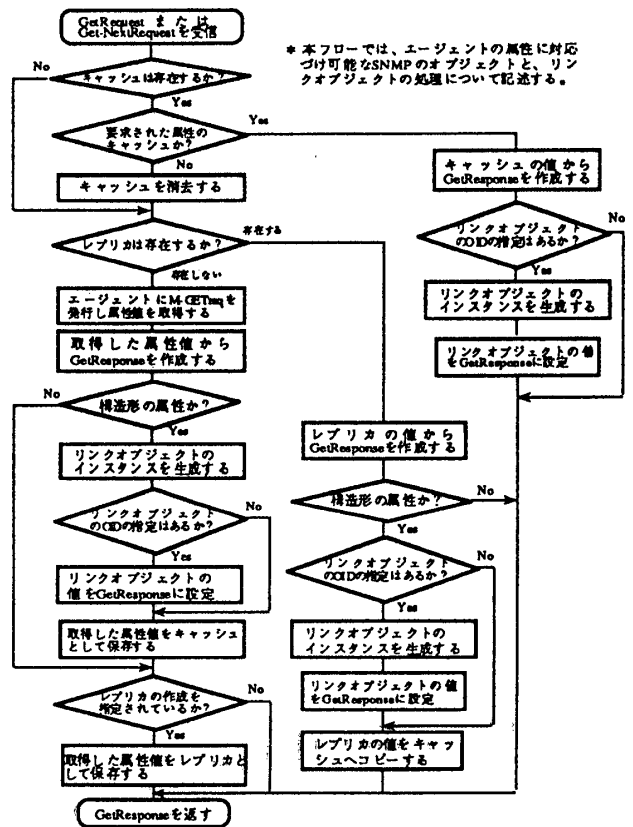


図4 GetRequest, Get-NextRequest の処理の流れ

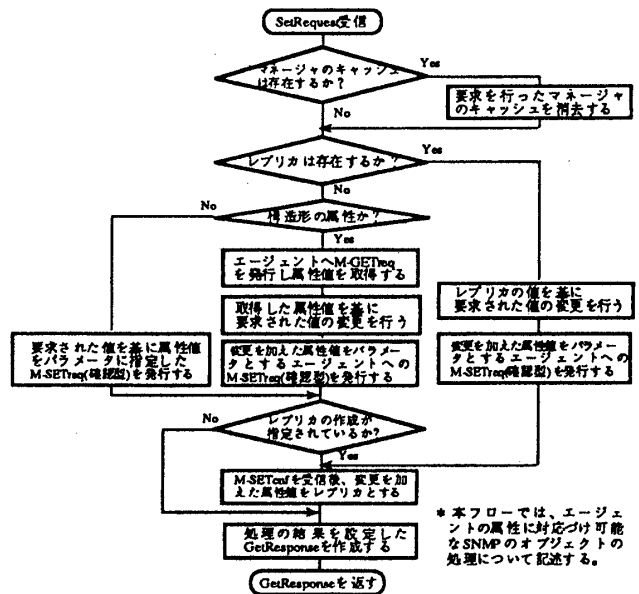


図5 SetRequest の処理の流れ

効率的な変換を実現できることを示した。最後に日頃ご指導頂く KDD 研究所浦野所長に感謝します。

参考文献

[1]:堀内,黒木,杉山,小花,鈴木"SNMP/OSI管理ゲートウェイの設計",情処第51回全大,1F-03,Sept. 1994.  
 [2]:NM Forum "Forum 030 : Translation of ISO/CCITT GDMO MIBs to Internet MIBs", Oct. 1993.