

## Meta Operation for SNMP Services の設計と実装

木田敦子<sup>1</sup> 遠山緑生<sup>2</sup> 船渡大地<sup>2</sup> 萩野達也<sup>1</sup>

<sup>1</sup>慶應義塾大学 環境情報学部

<sup>2</sup>慶應義塾大学 大学院 政策・メディア研究科

SNMP(Simple Network Management Protocol) は、均一かつ小規模なネットワークを想定して標準化されたネットワーク管理プロトコルである。しかし、ネットワーク構成は巨大・複雑化の傾向にあり、次世代のネットワークはSNMPによる管理が困難になると予測される。次世代ネットワーク管理へのアプローチの中でCORBA等の分散オブジェクト技術が、現在注目されている。本稿では、分散オブジェクト技術HORBを用いたネットワーク管理モデルMOSS(Meta Operation for SNMP Services)を提案する。MOSSは、クライアント、エージェント、管理機器から構成され、クライアント、エージェント間はHORBプロトコルを用いて通信を行なう。また、エージェントは複数のSNMPポーリングをエージェントの存在するローカルセグメントのみで行なう。監視対象機器からの応答も、エージェント内で保存し、ポーリングの終了時にクライアントへ監視データの転送をする。この方式により、SNMPのポーリングを行なう方式に比べてネットワーク全体に派生するトラフィックの量が削減される。さらにプロトタイプの評価を行い、提案の有効性を示す。

## Design and Implementation of Meta Operation for SNMP Services

Atsuko Kida<sup>1</sup> Norio Touyama<sup>2</sup> Daichi Funato<sup>2</sup> Tatsuya Hagino<sup>1</sup>

<sup>1</sup>Environmental Information, Keio University

<sup>2</sup>Graduate school of Media and Governance, Keio University

Simple Network Management Protocol (SNMP) is the standard network management protocol in the Internet. SNMP is designed to be suitable for homogeneous and small networks. Since networks have greatly changed and become more complicated and heterogeneous in the last decade, the methods used for managing the homogeneous networks of the past are no longer applicable. To solve this problem of managing heterogeneous networks, we use distributed object technology. In this paper, we propose a new network management model called MOSS (Meta Operation for SNMP Services) implemented with HORB distributed object technology. MOSS network management system consists of clients, agents and managed resources. A MOSS agent collects data from managed resources in the same network segment using SNMP polling and keeps their status in the agent. A MOSS client communicates with a MOSS agent using HORB to get the status. Since clients do not need to do multiple SNMP polling directly to managed resources, the overall network traffic can be reduced. We have evaluated the performance of our prototype system, and confirmed its validity.

# 1 はじめに

SNMP[1]はネットワーク構成が現在に比べて均一かつ小規模であった1980年代後半に提案され、1990年にRFC1157によって標準化されたネットワーク管理プロトコルである。さらに1993年には、SNMPバージョン2[2]と呼ばれるセキュリティと他ネットワーク管理システム(OSI)の親和性も考慮した仕様が定義された。しかし、LANの普及に伴い、標準化された当時に比べ、想定するネットワークの形態は変化し、現在ネットワークは複雑、巨大化の傾向にある。

このような背景から、次世代のネットワークはさらに管理が困難になると予測できる。次世代のネットワーク管理手法についてはいくつかのアプローチが現在提案されている。WBEM[5]は独自のスキーマ定義に従い、HMMP(HyperMedia Management Protocol)と、分散オブジェクト技術DCOM(Distributed Component Object Model)を用いたSNMPの遠隔操作によるネットワーク管理モデルを提案している。JIDM(Joint Inter Domain Management)は、分散オブジェクト技術CORBA[6]にTMN(Telecommunication Management Network)とSNMPの収容方式IDL(Interface Definition Language)定義を規定し[4]、その提案方式の有効性を実装を通して示している[8]。

本稿では、このような分散オブジェクト技術によるSNMPの遠隔操作が行なわれる次世代ネットワークを想定し、ネットワーク管理モデルMOSS(Meta Operation for SNMP Services)の提案を行なう。MOSSネットワーク管理モデルは、クライアント、エージェント、管理対象機器により構成されている。また、MOSSは分散オブジェクト技術HORB[3]を用いて実装したネットワーク管理モデルである。

MOSSではエージェントが存在するローカルセグメント内で複数のSNMPポーリングを管理対象機器に対して行なう。管理対象機器からの応答も、エージェント内で一旦保持し、ポーリングの終了時にクライアントへまとめてデータを送信する形式をとり、クライアントとエージェント間のトラフィックを削減する。またローカルセグメント内で提案したネットワーク管理モデルに基づいたプロトタイプ実装の実験を行なった。

その結果、MOSSはエージェント内でデータを保持する機能を用いた操作を利用すると、管理プロトコルによるトラフィックをSNMPの約2分の1から3分の1に抑えることができ、提案の有効性を示した。

本稿では、第2章においてSNMPの通信方式の概要とともに巨大、複雑なネットワークでのSNMPの問題点を挙げ、MOSSによる解決法の概略について述べる。第3章では、MOSSの設計と実装を通し、

SNMPの問題点の解決について述べる。また第4章では実装したプロトタイプに対して評価を行なう。

## 2 ネットワーク管理モデル

### 2.1 SNMPv1

図1にSNMPv1(バージョン1)のネットワーク管理モデルを示す。SNMPv1の通信モデルは、ポーリングによる同期型(図1(1))とトラップによる非同期型(図1(2))の二種類がある。

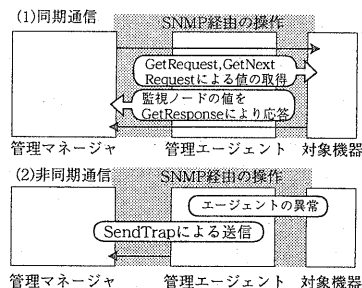


図1: SNMPv1 ネットワーク管理モデル

#### 2.1.1 ポーリングによる同期通信

図1中の(1)において、管理マネージャは、管理エージェントに対して問い合わせを行ない、機器内の対象オブジェクトの値を要求する<sup>1</sup>。管理エージェントは、対象オブジェクトの値を応答する<sup>2</sup>。

また、定期的に監視を行なう場合は、管理マネージャはSNMPを管理エージェントに対して一定時間毎にポーリングし、管理エージェントの応答を取得することで監視を行なう。

#### 2.1.2 トラップによる非同期通信

図1の(2)において、管理エージェントは対象機器に異常な事態が生じた時に管理マネージャに対し非同期に通知を行なう。

#### 2.1.3 問題点

ネットワーク構成が大規模で複雑な次世代ネットワークに移行する時に想定されるSNMPによる管理方式の問題点を次に挙げる。

<sup>1</sup>SNMPではGetRequest, GetNextRequestと呼ばれる。

<sup>2</sup>SNMPではGetResponseと呼ばれる。

- アクセス制限

SNMP は認証機構の弱いプロトコルである。管理マネージャと管理エージェントが異なるネットワークセグメントに存在する場合には、管理マネージャからの SNMP 経由の操作をファイアウォールの設定で禁止している場合がある。

- トラフィック増加

対象機器数が多いネットワーク内で定期的にポーリングを行なうと、ネットワークトラフィック全体に占める管理プロトコルのオーバーヘッドが高くなる。

- トラップ定義の記述の難しさ

対象機器の情報には頻繁な参照を必要としないが重要な情報がある。そのような情報は管理エージェントの非同期通知(トラップ)で情報の取得を行なうべきである。しかし対象機器への非同期通知情報定義の実装には SNMP と ASN.1(Abstract Syntax Notation) の知識と理解が必要である。

## 2.2 MOSS

本稿では分散オブジェクト技術 HORB を用いたネットワーク管理モデル MOSS の提案を行なう。MOSS は前述の SNMP ネットワーク管理に於いて挙げられた 3 つの問題に対し、以下のアプローチにより解決を行なう。図 2 にネットワーク管理モデル MOSS の概略図を示す。MOSS はクライアント、エージェント、監視対象機器から構成されるネットワーク管理モデルである。

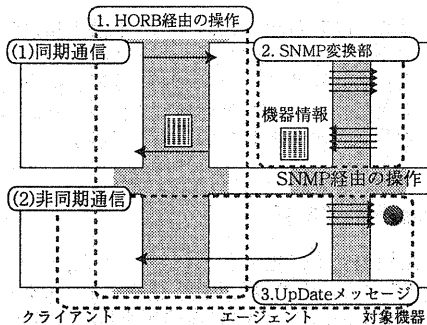


図 2: MOSS ネットワーク管理モデル

### 1. 通信モデルによるアクセス制限の解除

MOSS は、クライアントからエージェントに対して、HORB プロトコルを使って通信を行なう。エージェントはクライアントから送られ

た対象機器管理情報をエージェント内で SNMP プロトコルに変換し、ポーリングを行なう。

SNMP では、管理マネージャと管理エージェントが異なるセグメントに存在すると、管理エージェント側でネットワークアクセス制限がかかり、SNMP の直接操作ができない場合がある。しかし、MOSS は HORB の認証機構を利用して、HORB プロトコルによる SNMP の遠隔操作を行なっているため、異なるセグメントに対しても SNMP 管理ができる。

### 2. SNMP 変換部によるトラフィック減少

エージェントに対して、複数対象機器の情報取得を設定した場合、SNMP のポーリングは、エージェントが存在するローカルセグメント内のみで行なわれる。この方式を採用することで、ネットワーク管理プロトコルによるトラフィックを、他のネットワークセグメントに派生することのない操作が可能になる。

### 3. UpDate メッセージによるトラップの拡張

SNMP では非同期通知の対象となっているオブジェクトは限定されており、他のオブジェクトへの拡張を行なうことが難しい。

MOSS では、エージェントが、ローカルセグメント内でポーリングを定期的に行い、状態が変化したことで UpDate メッセージをクライアントに通知することができ、すべてのオブジェクトに対してトラップを拡張することが容易にできる。

## 3 MOSS の設計と実装

### 3.1 通信モデル

MOSS は、JAVA 言語による分散オブジェクト技術 HORB[3] を用いた通信を利用する。クライアントからネットワーク管理を行うためには、対象機器管理情報の記述されたオブジェクトを HORB プロトコルを用いてエージェントに送信する。

つまり、クライアントは、HORB を介して間接的に SNMP のポーリングを行なう形式をとる。また、MOSS は HORB の認証機構を用いてクライアントとエージェントが通信を行なうため、異なるセグメントからの SNMP による遠隔操作を禁止しているネットワークに対する管理を実現する。

MOSS の通信モデルを図 3 に示す。

1. クライアントは HORB サーバーが実行されているホストを URL 名で指定し、エージェントの代理オブジェクトを生成する。このときエージェント側ではクライアントに対してアクセス認証を行なう。



表 1: 監視機器管理情報のフォーマット

名前	主な機能・属性
Target	監視ノードホスト名
ObjectID	監視オブジェクト ID
Opname	オペレーション名
MIBName	MIB 名
READCOM	read community
WRITECOM	write community

### 3.2.2 SNMP ポーリングメソッドの実行

エージェントはローカル環境での複数ノードに対する SNMP のポーリングを実現するとともに、いくつかの SNMP ポーリングに対して拡張をおこなった。表 2 に、エージェント内 SNMP ポーリングメソッドの種類について示す。

表 2: SNMP ポーリングメソッド

名前	主な機能・属性
GetRequest	オブジェクトの値を取得
GetNextRequest	MIB の辞書順番の次の値を取得
GetWalk	探索点の終点までの値を全て取得
GetTable	テーブルの値を全て取得

表中の GetRequest, GetNextRequest は SNMP 標準のメッセージである。この他にも標準のメッセージとして SetRequest(MIB 内のオブジェクトの値を変更する)が挙げられるが、これについては MOSS はポーリングによるネットワーク監視に重点を置き提案したものであるため、提供は行っていない。

#### ● GetRequest

GetRequest は指定された ObjectID の取得を行なう。複数オブジェクトの属性値取得を目的として変数のバインディングも行うために拡張を行なった。

#### ● GetNextRequest

GetNextRequest は指定された ObjectID の MIB ツリー構造の次の値の取得を行なう。

#### ● GetWalk

GetNextRequest は MIB ツリーの最下層まで探索を行なうと、GetResponse を要求時と異なる ObjectID で応答する性質をもつ。この性質を利用し、GetWalk メソッドは、MIB 内の ObjectID の探索を Info オブジェクトで指定したオブジェクト ID をノードの端点として、MOSS サーバ内で GetNextRequest を再帰的

に呼び出す。そして、MIB の辞書的順序のもっとも深い ObjectID までの値を全て返す。

#### ● GetTable

GetTable メソッドは設定された MIB ファイルの Syntax が Table 型である場合に利用する。GetTable メソッドは Info オブジェクトの ObjectID を Table 型のノードの端点として、MOSS サーバ内で、表のカラム数を確認し、そのエントリ中の全ての値の取得を行なう。

## 3.3 Update メッセージ

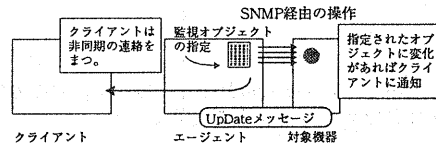


図 5: Update メッセージ

Update メッセージでは、クライアントは能動的な操作を行わず、エージェントがローカルセグメント内で指定された対象機器へのポーリングを定期的に行なう。値が変化した場合に、エージェントはクライアントに向けて Update メッセージの通知を行なう。Update メッセージには、HORB の非同期通信機構を用いる。

## 4 プロトタイプの評価

### 4.1 プロトタイプの実装

第 3 章と第 2 章で提案、設計した、MOSS ネットワーク管理モデルのプロトタイプを実装した。実装は、FreeBSD2.2.7 をプラットフォームとし、jdk1.1.7, HORB version 1.3.b1 を用いた。

### 4.2 プロトタイプの実験

実験はプライベートネットワーク内で行った。実験環境を図 6 に示す。単純な SNMP のネットワーク管理との比較対象として、ucd-snmp3.5 パッケージを用いて MOSS と比較する。SNMP では、管理マネージャから対象機器間 (図 6(A)), MOSS ではクライアントからエージェント間 (図 6(B)) の転送パケット数を測定する。

### 4.3 実験結果と考察

表 3 に、SNMP と MOSS ネットワーク管理プロトコル転送量の比較を、図 7 にはグラフ化した実験

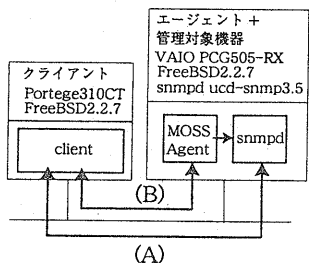


図 6: 実験環境

結果を示す。system.sysDescr.0 は (a) と (b) の実験で使われたオブジェクト名である。

表 3: 転送量の比較

測定項目	SNMP	MOSS
system.sysDescr.0 の属性値を		
(a) 1 個取得	109	179
(b) 変数 binding を用いて 10 個取得	460	663
system 以下の 16 個の属性値を	2019	1289
(c) GetNextRequest により取得	(880)	
if 以下の 102 個の属性値を	12029	3643
(d) GetNextRequest により取得	(5930)	

単位 (byte)

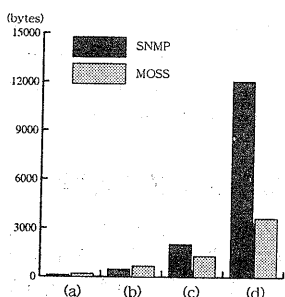


図 7: 実験結果

表 3 の監視対象のオブジェクト数が少ない場合 (a), GetRequest による変数バインディング<sup>2</sup>を行なった

<sup>2</sup>SNMP では、複数オブジェクトの属性値取得には変数バインディング機構を用いる。

場合 (b) においては、MOSS 通信モデルの HORB プロトコルによる管理情報オブジェクトの初期化の際のオーバーヘッドの数字が高くなっていると考えられる。

表 3 の (c),(d) の実験は、エージェント内で対象機器に対し、複数のポーリングを行なった場合に、SNMP の通信パケット量に比べて、(c) では約 2 分の 1, (d) では約 3 分の 1 に転送量の減少がみられた。表中の括弧内の数字は、全体の通信量内における GetNextRequest のポーリングの転送量である。MOSS はこれらのポーリングを、エージェントのいるローカルセグメント内で処理することで転送量の削減を可能にした。

## 5 おわりに

本稿では、分散オブジェクト技術 HORB を用いたネットワーク管理モデル MOSS の提案、プロトタイプ実装を行なった。プロトタイプ実装のネットワークトラフィック量の評価を通して提案方式の有効性を示した。

## 参考文献

- [1] Case, J., et al.: "A Simple Network Management Protocol(SNMP)", IETF, RFC 1157, 1990
- [2] Case, J., et al.: "Introduction to version 2 of the Internet-standard Network Management Framework", IETF, RFC 1441, 1993
- [3] Hirano, S.: "HORB: Distributed Execution of Java Programs", Worldwide Computing and Its Applications '97 (WWCA97), 1997
- [4] JIDMXoJIDM working group.: "Inter-Domain Management Specification Translation", X/Open and NMF, Ed. Open Group, 1997
- [5] J. Patrick Thompson.: "Web-Based Enterprise Management Architecture", IEEE Communication Magazine vol.36, pp80-81, 1998
- [6] Object Management Group.: "The Common Object Request Broker: Architecture and Specification Rev 2.0", 1995
- [7] Rose M., et al.: "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", IETF, RFC 1213, 1991
- [8] 堀内, 吉原, 小花.: 分散オブジェクトを用いたネットワーク管理に於ける TMN/SNMP 収容方式, 情報処理学会論文誌, Vol.40, No.1, pp.103-112, 1999