

分散協調管理プラットフォームの実現

犬束 敏信* 藤崎 智宏* 蔭山 克禎†

NTT 情報流通プラットフォーム研究所*
NTT 西日本 研究開発センタ†

概要

本稿では、筆者らの提案する分散協調管理プラットフォーム MII を用いた、インターネットの協調管理を支援するシステムの実装例を報告する。管理アプリケーション間での管理情報共有システム、管理パケットの自動集約システム、および、既存の IPv4 用 NMS を用いた、IPv6 ネットワーク管理の 3 システムについて報告する。これらの機能実現に於いては、管理アプリケーションとして市販のものを用いており、この管理アプリケーション自体には全く手を加えることなく、プラグ・アンド・プレイでの協調管理支援を実現している。また、これら実現されたシステムは、実際の国際規模でのネットワークの管理に用いられている。

An implementation of the platform for distributed cooperative management.

Toshinobu INUZUKA*, Tomohiro FUJISAKI*, Katuyoshi KAGEYAMA†

NTT, Information Sharing Platform Laboratories*
NTT West Corporation, Research and Development Center†

Abstract

This paper discusses a prototype implementation of MII. MII (Management Information Interchange) is a platform that enables the different network management applications to cooperate. We report 3 types of prototype implementations. These prototype applications enable to reduce load of managed network devices by aggregating management packets, and to share management result/status among management applications. We also describe IPv6 network management by MII and IPv4 NMS. These prototype applications are in practical use on Internet backbone service.

1. はじめに

インターネットに特徴的な管理体制として、複数の組織間で管理領域がオーバーラップするような複雑な形態がある。我々は、このような管理形態を「協調管理」と呼んでいる。既存の管理システムによる協調管理では、システムが前提としている集中型や階層型の管理モデルとは異なることから、いくつかの問題が発生している。例えば、複数の管理アプリケーションが独立に管理パケットを送信することにより管理対象機器の負荷が増大することや、あるいは

ン間)での作業の分担、作業経過の引き継ぎがしにくい、といったものである。これらの諸問題を解決するために、我々は分散協調管理プラットフォーム MII (Management Information Interchange)を提案している[1]。

本稿では、我々の提案する MII プラットフォームを用いた、管理アプリケーション間での管理情報共有システム、および管理パケットの自動集約システムの実装例を報告する。また、本プラットフォームの応用例として、既存の IPv4 向け NMS を用いた、IPv6 ネットワークの管理についても報告する。これらの機能の実現に於いて、管理アプリケーションには、独自実

装を用いたのではなく、市販の汎用 NMS を用いた。また、管理アプリケーション自体には全く手を加えることの無いプラグ・アンド・プレイでの協調管理支援を実現している。これら実現されたシステムは、実際の国際規模でのネットワークの管理に利用されている。

2. 協調管理プラットフォーム MII

2.1 協調管理

本稿では、2つ以上の管理主体がネットワークの一部を交互に、あるいは協調して管理するような管理形態を「協調管理」と定義する。一般に管理とは監視と制御の双方を指している。このような協調管理が必要な場面としては、

- ・ 昼夜間交代での管理
- ・ 外部組織への管理のアウトソーシング
- ・ 一次監視部門と、エキスパート部門の分業などが想定される。

協調管理では、必ずしも、協調する組織が同一の管理ポリシー、スキル、ツールなどを有しているとは限らず、むしろ不均一であることも多い。

また、単一の組織で分担管理する場合は、ポリシーやスキル等が統一されていることから、単一のシステムを利用したり、あるいは協調を念頭において、独自システムを開発したりすることが可能である(実際には、膨大な開発費用が必要であり、また、インターネットの変化に追従するには、開発したシステム維持管理費が大きくなり、リスクを伴う)。しかし組織が異なれば、独自システムを開発し、これを一斉導入することはほぼ不可能である。

我々のプラットフォームは、このような異組織間、異システム間の協調管理を念頭におき、設計されている。具体的な実装法については、3節で述べる。

2.2 既存 NMS での管理モデル

インターネットでは標準的管理プロトコル SNMP(Simple Network Management Protocol) [2]を用いた管理が一般的となっている。SNMPでは、トラップおよびポーリングという手法により機器の監視、制御を可能としている。トラップとは、機器の状態が予め決められた状態に変化したときに、ネットワーク管理システム(以下、NMS)へと通知するものである。また、

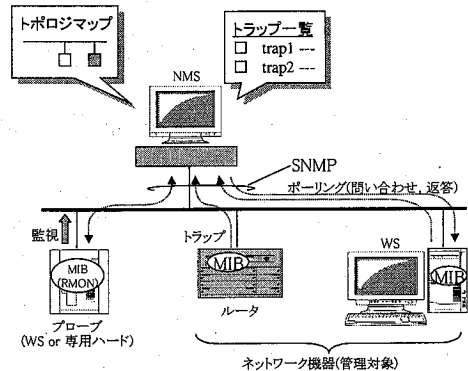


図1 集中型管理システム

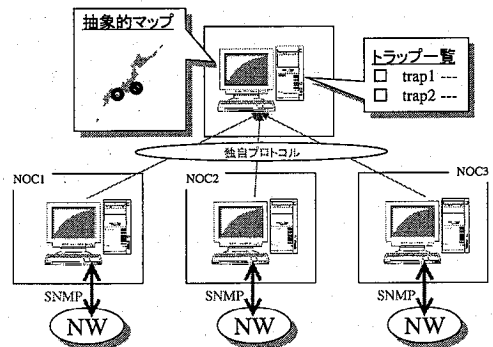


図2 階層型管理システム

ポーリングとは NMS 側から一定間隔で機器から情報を取得するものである。集中型の管理では、1つの NMS でトラップを受け、またポーリングを行う(図1)。階層型の管理では、複数の NMS の情報を束ねる NMS を設け、統合管理を行う(図2)。これら管理システムはいずれも、

- ・ ネットワーク機器からの情報の収集
 - ・ 収集された情報の解析、表示
 - ・ オペレータによる機器制御の支援
- といった基本機能を備えている。

2.3 協調管理プラットフォーム

協調管理の実現には、これら基本機能の他に、

- ・ 排他制御
- ・ 管理状態の相互参照や引継ぎ
- ・ 管理情報の統一
- ・ 管理負荷の低減

といった機能を実現する必要がある。このような協調管理特有の機能の実現には、独自システムを開発するよりも、デファクトスタンダー

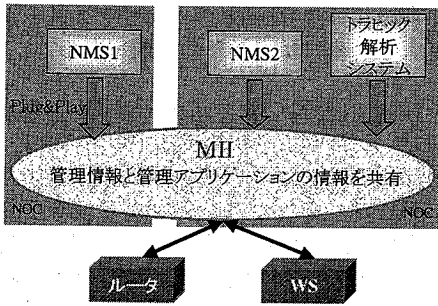


図 3 協調管理プラットフォーム MII

ドとなっている製品群を組み合わせることで管理できる方が望ましい。しかし、現状では、協調管理のための標準というものはまだ確立されていないため、これら製品間の連携をすることが困難である。

MIIは、このようなジレンマを解決する手段を提供する。集中型(階層型)管理モデルでは、NMSと機器が直接情報の交換をするが、MIIプラットフォームでは、NMSと機器の仲立ちをすることにより、管理情報の共有を実現する。また、MIIを導入する際にも、既存のNMS等の管理アプリケーションに全く手を加えることの無い、プラグ・アンド・プレイでの協調管理を実現している。

3. 協調管理アプリケーションの実装

3.1 管理情報同期アプリケーション

協調管理のための要件の1つに、管理情報の共有が挙げられる。すなわち管理アプリケーション(NMS)が内部に持っている管理情報と、別の管理アプリケーションの持つ情報との同期を取る機能である。ここで管理情報の同期がとれているとは、

- ・ MIB やトラップの意味付けの整合がとれている
- ・ 機器から取得した情報、および障害への対処状況の整合がとれている

を満足している状態を意味する。MIIでは図4の構成により管理情報の同期を実現した。

同期エージェント

SNMPトラップの受信によるNMSの管理イベント登録の発生を監視し、新規同期オブジェクトを同期マネージャに登録する。また、MIB定義の変更を監視する。

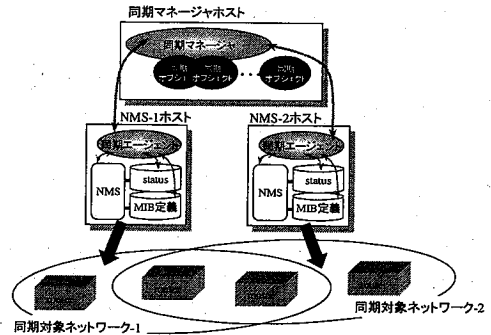


図 4 管理情報同期アプリケーション

同期オブジェクト

同期すべき情報をオブジェクト化したものである。同期エージェントによって生成され、同期マネージャに登録管理される。MIB定義、管理イベントなど。

同期マネージャ

同期を司るマネージャである。同期オブジェクトを管理し、同期エージェントからの登録、変更を受け、同期が必要な同期エージェントに配信する。

今回の実装では、同期するNMSとしてHP社のOpenView Network Node Manager[3](以下、NNM)を対象とした。この場合、同期すべき情報は、イベントブラウザ上での操作情報や、ロードしたMIB定義情報である。NNMでは、Ver5とVer6では、内部の情報表現方法が変わっているが、MIIでこの差分を吸収することによって、異版間での情報共有を実現している。

本アプリケーションは、日米間の時差を利用した、国際バックボーンでの管理に用いられている。新しい機器の追加時や、昼夜間の管理交代時の状態引継ぎに用いられる(図5)。

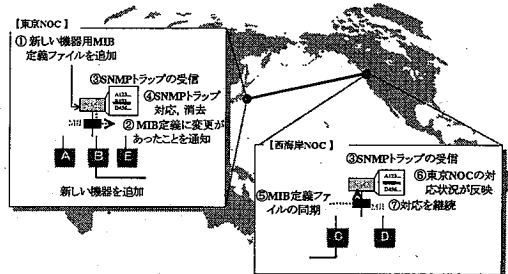


図 5 日米間での協調管理への適用

3.2 管理パケット集約アプリケーション

協調管理を行う場合、管理対象機器は複数のNMSにより管理されるため、それぞれのNMSからSNMPによるリクエストを送信されることになる。このため機器はその都度リプライを返す必要があり、機器の負荷がNMSの台数分だけ増加することになる。一部のNMSでは、Proxy機能によりこのような管理パケットの集約が可能であるが、この機能に未対応なNMSも存在する。本アプリケーションはこのようなNMSによって異なる機能に依存せず、キャッシュ機構を用い、重複した管理パケットを集約することにより、機器の負荷低減を実現している。

本システムの動作原理を図6に示す。最初にリクエストが投げられる場合には、次の手順をとる。

- (1) NMS1 は機器に対しリクエストを送信する。
- (2) MII ホストはこのパケットを塞き止める（一般的なパケットフィルタリング機能を用いる）と同時に、データ収集エージェントはそのパケットを取り込み解析する。
- (3) 解析を行った結果を元に、データ管理エージェントにデータが保持されているか問い合わせを行う。
- (4) データ管理エージェントは、要求された値がキャッシュされているかどうかを判断しデータ管理エージェントに回答を返す。最初の要求なので、値が無い旨が返される。
- (5) データ収集エージェントは、機器に対してリクエストを送信する。
- (6) 機器はデータ収集エージェントに対してリプライを返す。
- (7) 返答された値は、データ管理エージェントに蓄積される。
- (8) データ収集エージェントは NMS1 にリプライを返す。

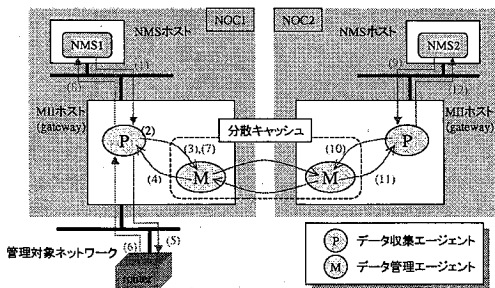


図6 管理パケットの集約

次に別のNMSから、同じ値へのリクエストが送信される場合には、以下の手順をとる。

- (9) NMS2 は機器に対しリクエストを送信する。
- (10) (2)および(3)と同様の処理を行う。
- (11) データ管理エージェントは、要求された値がキャッシュされているかどうかを判断しデータ管理エージェントに回答を返す。2度目の要求なので、キャッシュ値が返る。
- (12) データ収集エージェントは NMS2 にリプライを返す。

これにより、機器へのアクセスが集約されることになる。或る機器を管理するNMSの数が多くなるほど、キャッシュの効果は大きくなる。

キャッシュ更新の戦略

上記のような機構によりMIB値をキャッシュする際に考慮すべき点が幾つかあるが、特に重要であると考えられるのは、キャッシュの更新のタイミングである。MIIシステムではキャッシュされている値の更新に関して、次のようなポリシーを導入した。すなわち、「同じ管理アプリケーションからの要求に対しては、二度は同じ値を返さない」というポリシーである。前述の例において、さらにNMS1が同じに対してリクエストを出したときには、データ管理エージェントは、その値はキャッシュされているが、(NMS1にとっては)古いものだと判断し、新たに機器から値を取得するという手順をとることになる。

キャッシュの配置

管理対象の規模に応じて、データ管理エージェントを分散して配置することが可能である。データ管理エージェントを分散配置し運用する際の検索の負荷と遅延を低減するために、キャッシュの管理責任という概念を導入した。すなわち、1つの機器の情報は、1つのエージェントが管理するというモデルである。エージェントが或る機器のMIB値を初めてキャッシュした時に、その機器の情報は全て同一のエージェントが責任を持って管理することになる。MIB値単位ではなく、機器単位に管理することにより、検索空間を小さくすることができる。

また、機器とエージェントの対応関係は、分割して管理させずに、全てのデータ管理エージェントがそれぞれ内部に持っている。これは、

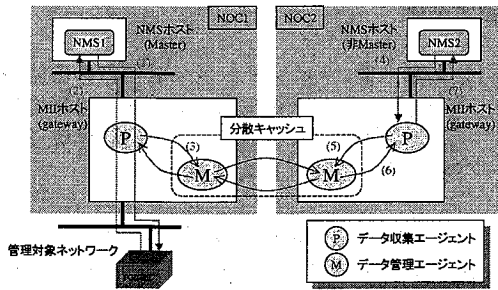


図 7 非対称な構成

機器とエージェントの関係が更新される頻度が、データを検索する頻度比べて極めて低いため、管理責任を得たことを全てのエージェントに通知するコストの方が、検索のたびに他のエージェントと通信するコストに比べて小さくてすむためである。

信頼性を高めるための非対称な構成

管理アプリケーションから送出されるパケットを MII ホストで塞ぎ止めているため、図 6 の例のような対称な構成では、MII 自体に障害が発生した場合には、全ての NMS は情報を収集できなくなる。このような状況を回避するために、非対称な構成による運用が望ましい。すなわち、図 7 に示すように、マスターとなる NMS を設定し、このマスター-NMS からのパケットは塞ぎ止めずに直接機器と通信させる構成である。この際、リプライのパケットだけを取り込み、キャッシュへと蓄積させることにより、非マスター-NMS がこのキャッシュされたデータを利用することが可能となる。

3.3 IPv6 ネットワーク管理アプリケーション

次世代プロトコルとして注目を浴びてきた IPv6 も商用アドレスブロックが配布され始めるなどのフェーズに至り、実験レベルではない商用ベースのネットワークの運用が開始され始めている。これにより IPv6 ネットワークの管理もこれまで以上に重要性が高まってくることが予想される。しかしながら、現状では、IPv6 ネットワークを管理できる製品はまだごく僅かである。

このような状況を踏まえて、IPv4 用に開発された NMS を用いて、IPv6 ネットワークの管理が可能なアプリケーションを実装した。

本アプリケーションの基本的概念は、パケット集約アプリケーションと同等なものであり、

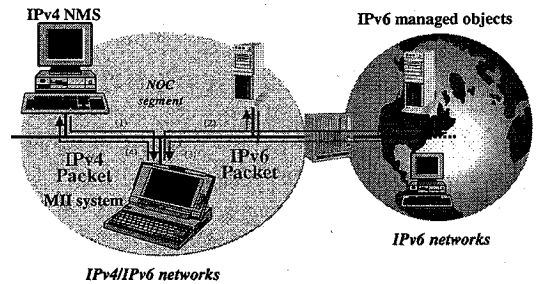


図 8 IPv4 NMS による IPv6 ネットワーク管理

管理アプリケーションから送出されるパケットを捕捉し、変換を行った後に MII から管理パケットを送出するという構成を取る。本システムの動作原理を図 8 に示す。

IPv4 専用開発された NMS では IPv6 のアドレスを理解できないために、管理対象機器 (IPv6) には、ダミーで IPv4 のアドレスを割り当てる。割り当てるといっても実際の機器にアドレスを設定する必要は無く、実際の IPv6 アドレスとダミーの IPv4 の対応表を MII アプリケーションに与えるだけである。

NMS と管理対象機器間の情報の伝達は、下記の手順により行われる。

- (1) IPv4 用の NMS から情報を取得した機器のダミーアドレス (IPv4) に向けて、管理パケットを送出する。
- (2) MII ホストでは、この管理パケットを捕捉し、内部に保持しているアドレスの対応表から機器の IPv6 アドレスを決定しこのアドレスに対して、管理パケットを送出する。
- (3) 機器は MII からの IPv6 での問い合わせに対して、IPv6 で返答する。
- (4) MII は機器からの返答を受け取り、リクエストを出した NMS へと IPv4 を用いてリプライを返す。

NMS ホストから MII ホストまで IPv4 パケットが届けば良いので、両者を同一セグメントに配置しておけば、管理対象ネットワーク内には、IPv4 パケットの流通は不要である。

このようにして、既存 NMS に手をくわえることなく、また管理対象のネットワーク機器への設定の変更も必要なく、IPv4 用 NMS を用いた IPv6 ネットワークの管理を実現した。

本アプリケーションは、NTT 情報流通プラットフォーム研究所が構築する IPv6 ネットワークの管理に用いられている。

4. おわりに

本稿では、我々が提案している分散協調管理プラットフォーム MII を用いた、アプリケーションの実装例 3 種類について報告した。今回のプロトタイプ実装では、基本的に Java による実装を行った（パケットのキャプチャ等の Java[4]での実装が困難な部分は C 言語で実装）。これは、動作環境できる限り制限を加えないこと、および分散オブジェクト通信を用いることによりエージェント間通信の信頼性を高めることを目的としたためである。

これらのプロトタイプ実装について、情報の同期アプリケーションは、国際バックボーンサービスの管理において実践的な評価を行っている。また、IPv6 ネットワーク管理アプリケーションは、NTT 情報流通プラットフォーム研究所が構築している IPv6 ネットワーク [5] の管理に試験運用されている。

管理パケット集約アプリケーションについては、現在研究所内において、性能などの評価を行っている。一つの指針として、キャッシュ値が遠隔のエージェントにあった場合の応答時間の遅延が、10 msec 程度であった。或るネットワークでの RTT (round-trip time) の計測結果として、東京～米国西海岸が 110 msec 程度、東京～米国東海岸が 160 msec 程度であったことを考慮すると、遅延に関しては現時点でも許容範囲のレベルであると考えられる。

キャッシュの戦略については、幾つかの検討すべき課題があると考えている。更新のポリシーを均一にしているが、管理情報にも、通過流量のように時々刻々と変化する情報や、装置の設置場所のような基本的に変化しない情報もあるので、適応的にポリシーを変化させれば、より高いキャッシュ効果が得られるはずである。また、管理責任は一度設定されると固定となっているが、機器と管理アプリケーションの位置関係から最適なロケーションへと遷移させることにより、さらに管理トラフィックを低減できると考えられる。

また、管理アプリケーション間での排他制御機構の実装についても今後の課題である。1つの機器を複数の管理アプリケーションが管理する協調管理においては、重要な機能であると考えられる。

参考文献

- [1] 犬東, 他. "インターネットにおける協調管理プラットフォームの提案", 情報処理学会分散システム運用技術研究会, 98-DSM-10, P.P.31-36, July 1998
- [2] Marshall T. Rose. "The Simple Book: An introduction to Internet management, 2nd Edition", PTR Prentice Hall, Inc., 1994.
- [3] Hewlett-Packard Company, "Network Node Manager", <http://www.openview.hp.com/products/nnm/>
- [4] Sun Microsystems, Inc., "The Source for Java™ Technology", <http://java.sun.com/>
- [5] NTT Information Sharing Platform Laboratories, "NTT IPv6 Home Page", <http://www.nttv6.net/>