

広域ネットワーク管理のための分散管理システムの構築

中井 智也† 泉 裕† 山口 英† 尾家 祐二‡

†奈良先端科学技術大学院大学

‡九州工業大学/奈良先端科学技術大学院大学

概要

インターネットの急速な普及にともない、ネットワークの管理にネットワーク管理システムを利用する組織が増えている。現在のネットワーク管理システムは、一般に SNMP (Simple Network Management Protocol) の機構を採用している。しかし既存のネットワーク管理システムでは、管理対象となる機器やアプリケーションの増加によって、管理トラフィックの増加や管理者へのオーバヘッドの増大などの問題が生じる。現在のインターネットアーキテクチャは新たなネットワーク機器を容易に接続できる。しかし一方では、管理対象となる機器の増加はその管理を著しく困難にしている。今後は管理システムの構造の拡張性を考慮したネットワーク分散管理システムの実現が求められる。

本研究では分散管理システム DAMM (Distributed Architecture using Multiple Managers) を提案し、さらに DAMM で用いる分散ネットワークマネージャの DNM (Distributed Network Manager) の設計および実装を行なった。

“ DAMM ” - A Distributed Management System for Wide Area Networks.

Tomoya Nakai† Yutaka Izumi† Suguru Yamaguchi† Yuji Oie†

†Nara Institute of Science and Technology

‡Kyushu Institute of Technology/Nara Institute of Science and Technology

Abstract

The SNMP-based network management systems are widely used in various environment. However, because of its design model, the SNMP-based systems have a major drawback, i.e, the scalability of the management system for the large-scale networks such as wide area networks or international enterprise Intranets. Therefore, the power of current network management systems is limited in terms of functions for adapting to the rapid growth of their managed networks.

In this paper, we address the problems in current network management systems, then we propose an distributed management system called “ DAMM ” (Distributed Architecture using Multiple Managers). The DAMM includes some DNM's (Distributed Network Managers) cooperating with each other in managed networks. Network Administrators can not only gather their network management informations effectively in using DNM but also add the DNM onto the managed networks easily. In this paper, the design and implementation of the DAMM system are also shown.

1 はじめに

近年のインターネットの急激な成長により、コンピュータ・コミュニケーションは社会的に浸透し広く利用されるようになった。さらにネットワーク利用形態も多様化し、遠隔医療や遠隔教育などの高度な応用も実現段階にある。このように社会におけるネットワークへの需要が高まるとともに、これまで以上にネットワークの安定運用が求められるようになってきた。しかし実際のネットワークの運用管理には豊富な技術力が必要であり、ネットワーク管理はごく少数のネットワーク管理者の多大な人的労力に依存しているのが現状である。

このような管理者の人的労力の軽減やネットワークの安定運用などの目的から、ネットワーク管理システムの導入が広く進められている。現在広く使われているネットワーク管理システムは、IETFによって標準化されたSNMP (Simple Network Management Protocol) と呼ばれる管理用プロトコルを使用している。SNMPを用いたネットワーク管理モデルはRemote Monitoring and Debuggingあるいはポーリング主導型と呼ばれる [1, 2]。このモデルでは、管理者が管理作業を行なう端末と、管理対象であるネットワーク上の機器やアプリケーションとの間で、SNMPを用いて管理情報を交換する。管理対象への問い合わせ(ポーリング)によって管理者は、管理対象の稼働状況を監視し、管理対象の持つ管理情報を変更することで管理対象の制御を行なう。

具体的な管理対象の監視や制御は、管理者側の端末内で稼働するマネージャ・プロセスと、管理対象となる機器内で稼働するエージェント・プロセスによって実現される。ポーリング主導型では、1つのマネージャが複数のエージェントを管理することが前提となっている。また複数のマネージャによってあるエージェントが管理される状況は、基本的に考慮されていない。

このモデルを、大規模なネットワークや複数のネットワークの管理に適用する場合、様々な問題が生じる。SNMPを用いたネットワーク管理では、1つのマネージャによる集中管理となってしまうため、大規模ネットワーク運用ではマネージャ自体の信頼性も考慮されなければならない。逆に複数のマネージャを用いて分散的に管理する場合には、マネージャ間の連携を可能とする新たな機構が必要となる。

本研究は複数のマネージャが連携して管理を行な

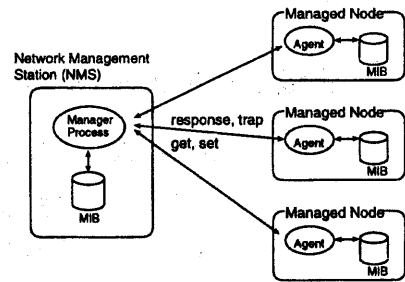


図 1: SNMP によるネットワーク管理

うようなネットワーク分散管理システムを構築することを目的とし、分散管理システムの構成要素となる分散ネットワークマネージャの設計と実装を行なう。

本稿では大規模ネットワークや複数のネットワークの管理を広域ネットワークの管理と記述する。

2 SNMP を用いたネットワーク管理

SNMP はネットワーク管理に広く用いられているプロトコルである [1, 2]。SNMP によるネットワーク管理モデルを図 1 に示す。管理者はネットワーク管理ステーション (NMS: Network Management Station) 上のマネージャ・プロセスを通して、ネットワーク上の様々な管理対象が持つ管理情報の取得や管理対象の制御が可能である。マネージャ・プロセスは管理対象となる機器内で稼働するエージェント・プロセスとの間で SNMP を用いた通信を行なう。SNMP で交換される管理情報は MIB (Management Information Base) と呼ばれるデータベース構造で管理されており、その構造や表現形式は標準化されている。本稿ではネットワーク管理を行なう人を管理者と呼び、NMS 上で起動されるマネージャ・プロセスをマネージャ、管理対象内で稼働するエージェント・プロセスをエージェントと呼ぶ。

SNMP によるネットワーク管理システムを用いて大規模ネットワークを管理する場合、以下のような問題が生じる。

管理ステーションへの負荷の集中

SNMP の管理モデルでは、単一のマネージャが複数のエージェントを管理する。大規模ネットワーク

では管理対象が多数あるため、マネージャの負荷はネットワークの規模に比例して増大する。

SNMPトラフィックの増大

多数の管理対象に対して定期的に情報取得する場合、情報取得の頻度が高いと SNMP トラフィックはネットワークに大きな負荷を与えてしまう。管理対象の増加によって NMS が接続しているネットワークの負荷はさらに深刻化する。

管理システムの信頼性

単一の NMS を用いて大規模ネットワークを管理する場合、管理者に対する負荷の分散、さらに管理システムの信頼性確保が重要である。SNMP の一元的な集中管理モデルでは、このような大規模ネットワークの管理システムに求められる技術的要件を満足させることができない。

以上のような問題点を解決するためにネットワーク分散管理システムが要求されており、現在までに以下のような技術が提案されている。

Manager-to-Manager MIB

Manager-to-Manager MIB(M2M MIB) [3] は、複数のマネージャ間の連携を目的としている。M2M MIB は、あるマネージャが他のマネージャに対して、指定した管理対象の監視を委託するために用いられる。M2M MIB では、委託する管理対象と MIB の項目、アラーム発生のためのしきい値、およびポーリング間隔等を指定することができる。委託された側のマネージャは、この M2M MIB の内容にしたがって監視を行なう。しかしながら、M2M MIB は管理対象の MIB を透過的に委託元のマネージャに提供するものではない。このため、単に監視だけが委託可能だが、管理対象の制御は直接行なわなければならない。また、M2M MIB の操作は SNMP を用いて行なうため、監視に必要な情報を一括して送ることはできない。このようなことから、M2M MIB の機構は分散管理には十分ではない。

分散オブジェクト指向プラットフォーム上の管理システムの構築

分散オブジェクトプラットフォーム上に管理システムを構築するという試みも行なわれている [4]。この試みでは M2M MIB に比べて柔軟な管理システム

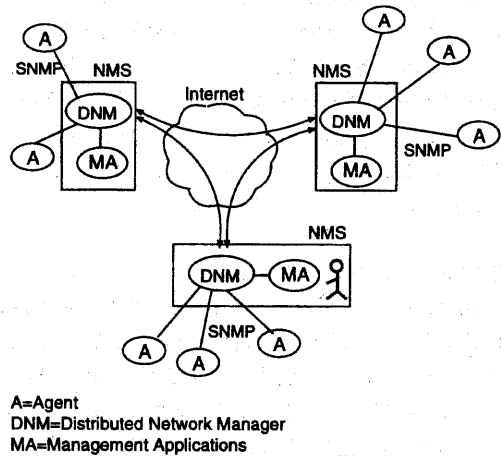


図 2: DAMM のモデル

の構築が可能であると考えられている。しかし、分散オブジェクトの名前解決にかかるオーバーヘッドなどのために実行速度が遅い点や、既存の管理システムとの協調が考慮されていないため実用性の問題がある。

本研究では上記の問題を解消する新たな分散管理システムの設計および実装を行なう。M2M MIB 等の分散管理システムとの相違を以下に述べる。

- MIB に制限されない、スクリプトによる委譲の方法を提供
- 既存の管理システムとの親和性

これらを考慮した分散管理システムを提案する。

3 提案システム

本稿で提案する分散管理システム DAMM(Distributed Architecture using Multiple Manager) のモデルを図 2 に示す。本システムは、複数の分散ネットワークマネージャ DNM (Distributed Network Manager) によって構成される。DNM は管理者や管理アプリケーションと管理対象とのインタフェースであり、DNM 間の連携によるネットワーク分散管理を目的としている。DNM 間の連携とは、各 DNM の管理するエージェントの持つ管理情報やエージェントを制御するための制御情報を DNM 間で転送す

ることを指す。DNM 間の通信には SNMP ではなく、新たに作成する DNM プロトコルを使用する。

このような各 DNM 間の連携は、管理者からは透過的であり管理者は複数の DNM の存在を意識する必要がない。管理するネットワーク内に分散する DNM によって得られる効果と、DNM の特徴について述べる。

管理システム構造の拡張性

DAMM では、個々の DNM が定期的に他の DNM の管理するエージェントの情報を得ることができる。従って新たに DNM を追加した場合も、その情報は他の DNM に伝わるので、管理対象の増加に応じて容易に DNM を追加できる。さらに DNM を介した監視の委譲には SNMP の機構を用いないので、M2M MIB のように委譲の内容が制限されない。

システムとネットワークの負荷分散

各 DNM は管理者の指示で管理するエージェントの持つ管理情報を収集し保持する。管理者の指示内容は管理情報の収集と管理対象の制御であり、複数の指示内容をスクリプトにして転送できる。したがって管理対象を一元管理する従来のモデルと比べて、システムやネットワークに与える負荷を軽減できる。

信頼性とセキュリティ保全

DNM 間の通信に TCP を用いることや暗号化や認証の機構を組み込むことで、高い安全性や通信到達性を確保できる。ただし、DNM とエージェント間はセキュリティ保全が確保されていると仮定し、DNM とエージェント間の通信には SNMP の機構を用いる。また本システムはあるマネージャが故障しても簡単に他のマネージャにより管理作業が代行できるため、システムとしての信頼性も高い。

既存の管理システムとの親和性

DAMM では SNMP による通信機構とアプリケーションインタフェースを提供する。NetView などのネットワーク管理システムは、エージェントとの通信に SNMP を用いるため、DAMM と従来のネットワーク管理システムとの親和性は高い。しかし、DAMM はスクリプト転送機能を持っており、スクリプト転送を DAMM に指示する場合はスクリプトエディタなど別のアプリケーションを提供する必要がある。既存のネットワーク管理システムにはスクリプト

設定を行なう機能が提供されていないので、管理者から DNM へのスクリプト転送には別のアプリケーションを提供し、DNM のスクリプト転送における評価に用いる。

4 システム設計

3 章でも述べたように、DAMM システムでは複数の DNM が分散して配置される。各 DNM 間は TCP をベースとした DNM プロトコルで通信を行なう。

本章では DAMM システムで用いられる DNM の機能と構成、および DNM 間の通信について述べる。

4.1 DNM の機能

各 DNM が連携するには、DNM がそれぞれ管理するエージェントの情報を交換し DNM 全体で共有する必要がある。エージェントの情報とはエージェントを含む管理対象のアドレスである。DNM は管理者によって指定されたエージェントのアドレスから、エージェントを管理する適切な DNM に要求を転送して、要求に対する処理をエージェントに実行させる。処理の結果がエージェントから DNM に報告されると、管理者と通信する DNM に結果を転送する。

エージェントから能動的に管理者へ通信する時、すなわち管理対象に障害が発生した場合は、エージェントから DNM に送られるトラップも適切な DNM に転送して管理者に通知する必要がある。

上記の動作を DNM の連携に必要な機能とし、次節で DNM のモジュール構成と各モジュールの役割について述べる。

4.2 モジュール構成

図 3 に DNM の構成を示す。DNM は以下に示す各モジュールから構成される。

SNMP 処理モジュール、MIB 処理モジュール

SNMP の通信機能を提供する。SNMP のデータグラムの生成や MIB に対するアクセスなどを行なう。

アプリケーション・インターフェース

NetView などのネットワーク管理アプリケーションとのインターフェースを提供する。

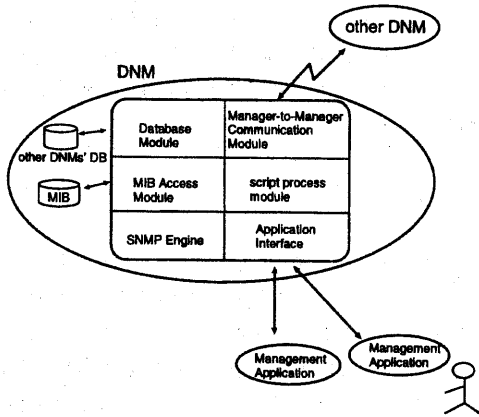


図 3: DNM のモジュール構成

スクリプト処理モジュール

一つの DNM から他の DNM に対して、複数の処理をまとめて転送する場合にスクリプトを使用する。スクリプトによって指定された内容を解析し、実行するためのモジュールである。

データベースモジュール

DNM が持つデータベースは、(1)DNM のアドレス、(2)DNM が管理するエージェントのアドレス、(3)トラップの配送先を設定するフラグから構成される。データベースモジュールは、このデータベースを用いて特定のエージェントのアドレスからエージェントを管理している DNM のアドレスを取得する。また、(3)のフラグを用いることでエージェントからのトラップの配送先を決定する。

マネージャ間通信モジュール

DNM 間の通信をサポートするモジュールである。DNM 間の通信については次節で詳細を述べる。

4.3 DNM 間の通信

DNM 間の通信には、各 DNM が管理するエージェントに関する情報の交換、監視の要求とそれに対する応答、トラップの転送の 4 種類がある。それぞれの通信について以下で述べる。

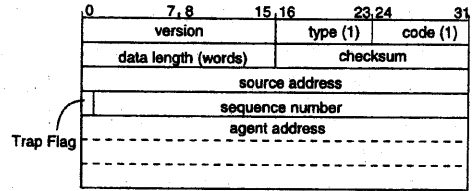


図 4: HELLO パケット

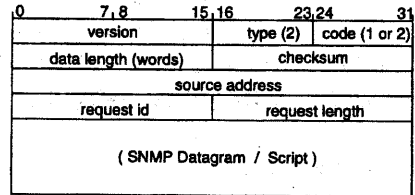


図 5: REQUEST パケット

4.3.1 各 DNM の管理するエージェントに関する情報の交換

DAMM では管理システムの追加を容易にするため、各 DNM が定期的に HELLO パケットを交換して、各 DNM が管理するエージェントを把握する。図 4は、HELLO パケットのフォーマットである。HELLO パケットは、ヘッダ中の type フィールドと code フィールドの値がそれぞれ 1 である。

HELLO パケットには、データとしてパケットのシーケンスナンバ、トラップ・フラグ、管理対象のエージェントのアドレス等が含まれる。

このパケットはデータベース・モジュールで処理され、各 DNM の持つデータベースに登録される。また HELLO パケットを受けとった DNM は HELLO ACK パケットを生成し送信する。

4.3.2 監視の要求および応答

REQUEST パケットは、管理ステーションからの監視要求を転送するために用いる。REQUEST の転送先の DNM のアドレスは、管理対象のエージェントのアドレスをキーとして、データベースから導出する。図 4は、REQUEST パケットのフォーマットである。REQUEST パケットは type フィールドが 2 であり、code フィールドは 1 もしくは 2 である。

要求の内容は、複数の要求からなるスクリプト型

のものと、単純な SNMP の操作のどちらかであり、code フィールドの値で区別する。code が 1 の場合はデータが通常の SNMP のデータグラムであることを示し、code が 2 の場合はデータがスクリプトであることを示す。

DNM は、受けとった REQUEST パケットの内容がスクリプトの場合にはスクリプトモジュールで解析・実行し、SNMP のデータグラムである場合には SNMP 処理部を通じて目的のエージェントに送信する。受けとった REQUEST に対する応答は、RESPONSE パケットを送信する。RESPONSE パケットには実行した REQUEST の要求 ID を付ける。

4.3.3 トラップの通知

トラップの通知は、DNM の持つデータベース中でトラップ・フラグが 1 になっている DNM に対してのみ送信される。

5 実装

現在、4章で述べた DNM の実装を進めている。実装用の言語としては C 言語を使用し、SNMP ライブラリには Carnegie Mellon 大学で開発された CMU SNMP version 2 パッケージに付属しているものを使用している。

現段階では、REQUEST と RESPONSE による SNMP 操作の転送を DNM 同士で行なう機構について実装済である。

6 今後の課題

今後は本稿で提案したモデルに基づき、HELLO パケットを交換する機構や TRAP の転送機構を完成させる。さらに、スクリプトの転送による効率的な管理要求の転送についても実装および逐次改良を行なう。

本システムの評価は、DNM によるマシンへの負荷およびネットワークトラフィックへの負荷を計測して従来の管理モデルと比較する。さらに管理対象や DNM の追加に伴なう上記の負荷の変化について検証する。

また、本システムでは DNM 間の通信の信頼性を向上するために TCP を使用していることや、DNM 間通信のセキュリティを強化するために SSLeay の

サポート等も検討していることなどから、SNMP を用いた場合よりも実行時間が遅くなる可能性がある。従って今後は実行速度に関する実験・評価も行なう予定である。

7 おわりに

本稿では、従来の SNMP を用いたネットワーク管理システムの持つ問題点について論じ、分散ネットワーク管理システムの必要性について述べた。

また既存の分散管理技術についても考察を行ない、ネットワーク分散管理システムに対する要求の分析と本研究で提案する分散管理システム DAMM で用いる分散ネットワークマネージャ DNM の設計を行なった。

DNM は DNM 間の定期的な情報の交換により、管理システム全体の変化に柔軟に対応しマネージャの追加・削除を容易に実現できる。

今後は本稿の設計で述べた DNM のモデルに基づいて実装を進める。セキュリティの強化についても検討する。

参考文献

- [1] J. Case et al.: Introduction to SNMPv2 (1993). Request For Comments 1441.
- [2] Marshall T. Rose: *The Simple Book*, Prentice Hall, 2nd edition (1994).
- [3] J. Case et al.: Manager-to-Manager Management Information Base (1993). Request For Comments 1451.
- [4] 藤崎智宏, 蔭山克禎, 浜田雅樹: 分散オブジェクトプラットフォーム上に構築したネットワーク管理システムの評価, 情報処理学会分散システム運用技術研究グループ研究会, Vol. 6, No. 8, pp. 37-42 (1997).