RFC 1157
SNMP (Simple Network Management Protocol)
分散システム特論レポート

木場 雄一 koba@csce.kyushu-u.ac.jp

2000 年 1 月 31 日

1 概要

SNMP (Simple Network Management Protocol) はインターネットワーク上のルータ管理のために開発されたプロトコルである。これはネットワーク上のデバイスからネットワーク管理情報を収集する手段を提供する。なお、通信には UDP を使用する。SNMP には広義と狭義があり、広義は MIB (Management Information Base) , SMI (Structure and identification of Management Information), SNMP からなるプロトコルスイートの集合を指す。

構造のモデル

- ネットワーク管理ステーション
  - エレメントを監視/制御する管理アプリケーションを実行する。
    → ステーション上で実際にデバイス管理を行なうソフトウェアをマネージャと呼ぶ。
- ネットワークエレメント
  - ルータ、ホスト、端末などがこれにあたる。そのエレメント内のデバイス管理をするソフトウェアをエージェントと呼ぶ。

SNMP はマネージャとエージェント間での通信をするために使われる。SNMP メッセージは単一のデータグラムで送信し、複数のデータグラムにまたがらない。

マネージャはエージェントにエレメントにおける知りたい情報を要求する。これを受け、エージェントはエレメント内の情報を格納している MIB を参照し、その形式に基づいて応答をマネージャに返す。

SNMP でやりとりする管理情報

SNMP では MIB と SMI で定義されたオブジェクトインスタンスを使用してマネージャ・エージェント間のやりとりをする。この表記法には ASN.1 (ISO が規定するデータ表記等のルール) を使用している。

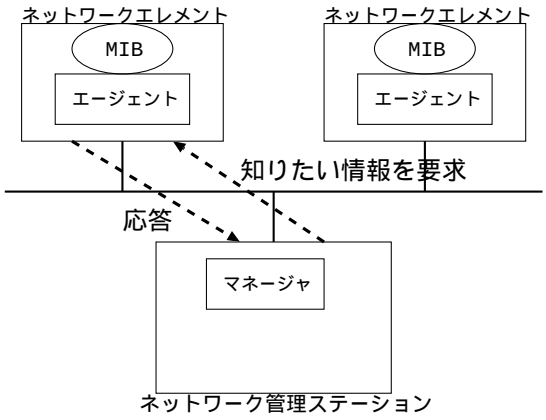

図 1: SNMP の構造モデル

MIB (Management Information Base)

RFC1156 で定義。ネットワークエレメントに置かれる、エレメント内の情報を保持するデータベースで、管理する項目を定義する。記述は SMI に基づいて行なわれる。

SMI (Structure and identification of Management Information)

RFC1155 で定義。MIB の中の変数を定義する。SMI は MIB に対するスキーム。

2 MIB

SNMP で用いられる MIB の形式について紹介。

2.1 オブジェクトのプロパティ

MIB には 4 つのプロパティが存在する。

- オブジェクトタイプ
- シンタックス
- アクセス
- ステータス

2.1.1 オブジェクトタイプ

オブジェクトの名前を定義するためのプロパティ。オブジェクトの名前には 2 種類あり、1 つはラベルで、もう 1 つは識別子。ラベルは単なる名前だが、識別子は木構造に従って表記する。

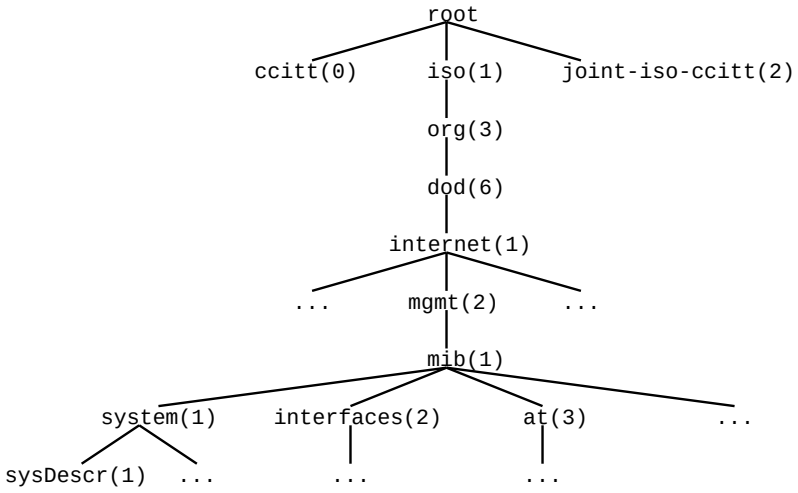

図 2: オブジェクト識別子

例えば、機器に関する説明情報を持つ sysDescr のオブジェクトクラスを記述する場合、図 **1** のツリーを root から下へ目的のオブジェクトまで辿っていき、

iso.org.dod.internet.mgmt.mib.system.sysDescr

または、

1.3.6.1.2.1.1.1

と表記する。

2.1.2 シンタックス

整数、文字列、オブジェクト識別子等のデータタイプを指定するためのプロパティ。

2.1.3 アクセス

特定のオブジェクトへのアクセスレベルを指定するもので、次の 4 つがある。

- READ-ONLY
- READ-WRITE
- WRITE-ONLY
- NOT ACCESSIBLE

何も記述されていない場合は、エージェントからの要求は破棄される。

2.1.4 ステータス

当該オブジェクトの実装条件を指定するプロパティ。

3 プロトコルの詳細

- プロトコル間のやりとり — ASN.1 の基本エンコードルールを使った UDP データグラムによってなされる。
- メッセージの構成 (以下の 3 つで構成)
  - version identifier : メッセージの整合性をチェックする
  - SNMP community name : エージェントにアクセスする時の認証の際、パスワードの役割をする
  - protocol data unit (PDU)
- trap メッセージ以外は UDP の 161 番ポートでメッセージを受け取る
  → trap メッセージは UDP 162 番ポート (プロセスを促進するため = 効率化)

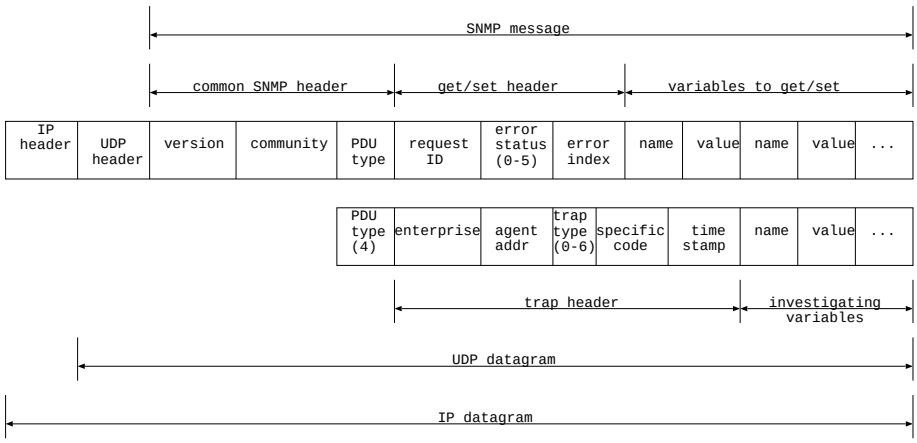

図 3: SNMP のフォーマット

3.1 PDU

PDU は次の 4 つで構成される。

- リクエスト ID : 未解決のリクエストの区別に使用。
  - SNMP アプリケーションのエントリは、入ってきた応答を未解決のリクエストに関連づけることが可能に。
  - 信頼性の低い場所でデータグラムのサービスがなされている場合、重複したメッセージを見分けることができる。
- エラーステータス : リクエストがなされている間に例外が発生した時、それを示すのに使用。非 0 のインスタンスで表す。
- エラーインデックス : 例外発生の原因になったリスト内の変数を示す。

  - noError(0) — 例外なし
  - tooBig(1) — エージェントがリプライを single SNMP メッセージに入れられなかった
  - noSuchName(2) — オペレーションが無効な値もしくはシンタックスを明記した
  - badValue(3) — マネージャーが read-only の変数に修正しようとした
  - genErr(5) — 上記以外のその他のエラー
- 変数バインディングリスト：ここに入る変数値は PDU においては無視される。

SNMP は 5 つの PDU を実装している。

1. GetRequest-PDU (マネージャ → エージェント)
   - 1 つの MIB 変数 (管理対象のオブジェクトのインスタンス) や MIB 変数のリストをエージェントに要求する。マネージャとエージェントのアドレスを必要とする。
2. GetNextRequest-PDU (マネージャ → エージェント)
   - MIB の中の、次の項目の変数をエージェントに要求する。これを使用することによって、マネージャは次々とエージェントの MIB を巡回することができる。
3. SetRequest-PDU (マネージャ → エージェント)
   - MIB 変数に新しい値 (インスタンス) を格納することを要求する。
4. GetResponse-PDU (エージェント → マネージャ)
   - 1. ~ 3. に対する応答。
5. Trap-PDU (エージェント → マネージャ)
   - あらかじめ定義した状況が発生したときに自動的に生成される
   - UDP ポート 192 番を使用 (1. ~ 4. は 191 番ポート)
   - 定義されている状況
     - coldStart Trap：エージェントが自身を初期化
     - warmStart Trap：エージェントが自身を再初期化
     - linkDown Trap：インターフェースがダウン
     - linkUP Trap：インターフェースが復活
     - authenticationFailure Trap：無効なコミュニティ名を受けた
     - egpNeighborLoss Trap：EGP ピアがダウン
     - enterpriseSpecific Trap：企業が作った固有のエラーコード

4 SNMP Version2

RFC1441 ~ 1452 で SNMPv2 が定義されている。

主な特徴

- MIB 構造の補強：SNMPv2 MIB, SNMPv2-M2M MIB を定義
- プロトコルの追加
- マネージャ間の通信が可能に
- セキュリティの強化
- PDU の追加
  - GetBulkRequest-PDU：バルクデータ転送を可能に
  - informRequest-PDU：マネージャ間の通信
  - Report-PDU：SNMP 製品情報の提供