9.5 ネットワーク監視

ここでは，ネットワーク監視に用いられるプロトコルである SNMP，及び，SNMP が取得する情報である MIB について解説する。

9.5.1 監視に用いられるプロトコル

ネットワーク監視システムによって違いはあるが，監視にはその目的に応じて様々な手法が用いられている。

表：監視の目的とその手法

監視の目的		手　法
稼働監視	ping 監視	**ping** パケットを定期的に送信し，ネットワーク層レベルでノードのダウンを検知する
	TCP 接続監視	監視対象のポートに TCP コネクションを確立し，トランスポート層レベルでノードのダウンを検知する
	アプリケーション接続監視	監視対象のポートに特定のコマンド（例えば，HTTP であれば HEAD リクエスト）を定期的に送信し，アプリケーション層レベルでノードのダウンを検知する
	SNMP 監視	SNMP パケットを定期的に送信し，死活確認を行う（SNMP で性能監視を実施していれば，機器のダウンも検知できる）
性能監視		SNMP パケットを定期的に送信し，特定の MIB 情報を取得する。取得したデータをネットワーク監視システム側で整形し，蓄積する
機器からのイベント通知		あらかじめ設定した条件を満たしたときに，機器側からネットワーク監視システム側に SNMP Trap パケットを送信し，イベントを通知する。 SNMP トラップは応答確認がないが，SNMP インフォームは応答確認がある。ただし，ホストが稼働し続けており，かつ，ネットワーク監視機器との接続性が確保されている必要がある
ネットワーク構成の管理（自動ディスカバリ）		**LLDP**（Link Layer Discovery Protocol）に対応したネットワーク機器を用いる。機器は，隣接機器に対し，自機器の ID（通常，MAC アドレス），インタフェースの情報を格納した LLDP フレームを送信する。機器は，LLDP によって得られた隣接機器の情報を自分の LLDP-MIB に保持する。機器間で相互に LLDP のやりとりをしていれば，各機器から LLDP-MIB を収集することで，ネットワーク構成を自動的に把握できる
		ローカルネットワークのブロードキャストアドレス（ディレクテッドブロードキャストアドレス）宛てに ping コマンドを発行すると，各ホスト宛に ARP 要求が次々に自動的に発行される。その結果，（ping 応答はないものの），ARP テーブルに実在機器の IP アドレスと MAC アドレスが記録される。この ARP テーブルの情報を用いて，ネットワーク構成を把握する

詳説

インタフェースを冗長化している場合，そのダウンを確実に検知したいときには，ping よりも SNMP を使用するとよい。ping 監視は IP パケットの到達性を確認するので，インタフェースに障害が発生しても，待機系のインタフェースに切り替わって IP アドレスが引き継がれるのであれば，ping は返ってくる。一方，SNMP では，インタフェースがアップしているかダウンしているかを保持する MIB 変数（ifOperStatus）が提供されているので，これをポーリングすればよい。また，インタフェースがダウンに変化したときにイベントを通知（linkDown trap）することもできる。

試験に出る

LLDP を用いた自動ディスカバリについて，令和3年午後Ⅰ問1，平成29年午後Ⅱ問1で出題された。
ARP テーブルの情報を用いた自動ディスカバリについて，平成22年午後Ⅱ問1で出題された。トランスポート層レベルでのネットワーク監視について，平成21年午後Ⅰ問3で出題された

試験に出る

SNMPのコミュニティについて，平成30年午後Ⅰ問2，平成23年午後Ⅰ問2で出題された

SNMP

ネットワーク監視システムは，監視する側（マネージャ）と監視される側（エージェント）から構成される。**SNMP** はネットワーク機器を監視するための標準的なプロトコルで，マネージャとエージェントでやり取りされるパケットのセットをサポートしている。

現在，SNMPのバージョンは1（SNMPv1），2（SNMPv2，SNMPv2cなど），3（SNMPv3）がある。SNMPv1とSNMPv2以降では，パケットフォーマットやトラップの種類が異なっている。

なお，バージョン2は，SNMPv1の機能拡張とセキュリティ強化を目標に策定されたものの，セキュリティ部分がRFC標準に採決されなかった（「歴史的」扱い）。その多くの機能はSNMPv3に引き継がれ，2002年にRFC標準となった。現在では，SNMPv1とSNMPv3の双方をサポートする機器が出回るようになっている。

関連RFC

RFC3411～3418（STD62）

エージェントが提供する情報を，**MIB**（Management Information Base）と呼ぶ。MIBの定義の仕方を定めたものをSMI（Structure of Management Information）と呼ぶ。

マネージャとエージェント間の通信形態は，次の三つに分類することができる。

- **マネージャとエージェント間の要求／応答**
  マネージャがエージェントに要求パケットを送信し，エージェントが応答パケットを返信する。MIB値の取得要求は，GetRequest，GetNextRequest，GetBulkRequestパケットを用い，一般に一定間隔（5分など）ごとに実施する。この動作を**ポーリング**という。なお，GetBulkRequestはバージョン2以降の規格である。
  MIB値の設定要求にはSetRequestパケットを用いる。それぞれの要求パケットに対し，エージェントが応答するときにはResponseパケットを用いる。
- **エージェントからマネージャへの通知**
  エージェントに対し，通知したいイベントを事前に登録しておき，当該イベントが発生したとき，エージェントからマネージャにTrapパケットを送信する。この動作を**トラップ**という。
  トラップの際，マネージャからエージェントには，何も返

詳説

GetRequestは，指定されたMIB変数を1個だけ取得する目的で用いる。GetNextRequestとGetBulkRequestは，MIBツリー上の指定された枝（サブツリー）に含まれる全てのMIB変数を取得する目的で用いる。GetNextRequestは，1回のやり取りで，取得したいMIB変数を1個指定する（MIB変数のOID値を辞書順に並べてみたとき，指定されたMIB変数の次に位置するものが取得対象となる）。GetBulkRequestは，1回のやり取りで，取得したいMIB変数を複数個指定できる

信されない。それゆえ，送信したTrapパケットが何らかの理由でマネージャに届かなかった場合，エージェントはその送信が失敗したことを知ることができないという問題がある。

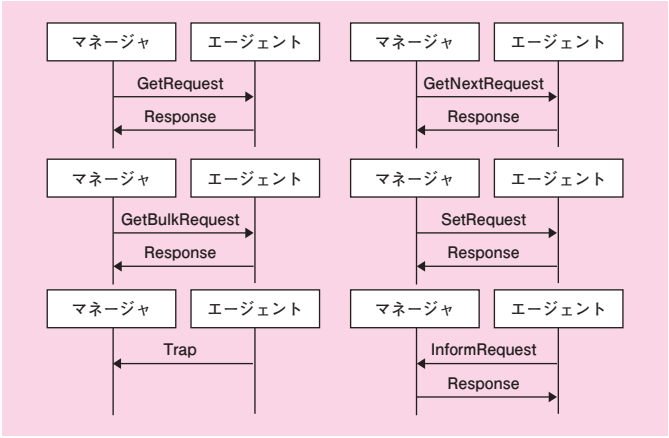
この問題を解決するため，後述するInformRequestがSNMPv2cで策定された。

- **エージェントからマネージャへの通知／応答**

これはSNMPv2以降でサポートされている。Trapと同様，エージェントからマネージャに情報を通知するときに用いられる。しかし，Trapとは異なり，マネージャはこの情報通知を受け取ると，エージェントに返信する。情報の通知はInformRequestパケットを用い，その返信はResponseパケットを用いる。

送信したInformRequestパケットが何らかの理由でマネージャに届かなかった場合，マネージャからResponseが返信されないので，エージェントはその送信が失敗したことを知ることができる。したがって，例えばInformRequestを再送するなど，何らかの対策を講ずることができる。

通常，マネージャには，取得するMIBオブジェクト及びポーリング間隔をエージェントごとにあらかじめ登録しておく。一方，エージェントには，通知するイベントとTrapパケットの送信先（通常



図：SNMPの通信形態

SNMP Trapについて令和元年午後Ⅰ問1，平成30年午後Ⅰ問2で出題された。Trapとは異なり，InformRequestはレスポンスが返ってくることについて，平成30年午後Ⅰ問2で出題された

SNMPは，トランスポート層プロトコルにUDPを使用する。使用するポート番号は，マネージャからエージェントへの要求が161番，エージェントからマネージャへの通知が162番である

はマネージャを指定）をあらかじめ登録しておく。ポーリングしたデータはマネージャに蓄積され，ネットワーク監視システムが統計処理やグラフ作成などの加工を施し，管理者にレポートする。

ネットワーク監視システムによっては，ポーリングしたデータの分析に基づき，自らイベントを発行して管理者に通知できるものがある。この場合，イベント通知のタイミングは，トラップのようにリアルタイムではなく，ポーリングのタイミングと同期してしまう。しかし，マネージャアプリケーションの機能により，トラップよりもきめ細かいイベントを定義できる，というメリットがある。

トラップで通知できるイベントには，次のようなものがある。ここでは，SNMPv1 の例を示す。

詳説

バージョン2以降は，MIB上でトラップを異なる仕方で定義している。バージョン1ではTRAP-TYPEとして定義されているが，バージョン2ではNOTIFICATION-TYPEとして定義され，多くのものが追加されている

表：トラップのイベント

種類（値）	内　容
coldStart（0）	エージェントが再起動された。全ての管理項目はリセットされる
warmStart（1）	エージェントが自ら再初期化を行った。管理に関する項目はリセットされない
linkDown（2）	機器のインタフェースがリンクダウンした。リンクダウンしたインタフェースが通知される
linkUp（3）	機器のインタフェースがリンクアップした。リンクアップしたインタフェースが通知される
authenticationFailure（4）	認証トラップ。コミュニティ名による認証に失敗した
egpNeighborLoss（5）	EGP（Exterior Gateway Protocol）の近接ルータとの通信が切断された
enterpriseSpecific（6）	ベンダの一般的なトラップであり，Enterprise 番号にはベンダ固有の番号（プライベート MIB のベンダ番号）が格納される。RMON エージェントから alarm を通知する場合，Enterprise 番号には RMON MIB サブツリーの OID（1.3.6.1.2.1.16）が格納される

SNMP は，セキュリティ機能をもっている。

SNMPv1 は，マネージャがエージェントの MIB 情報にアクセスする際，エージェント側でマネージャを認証する。このために用いられるのが，**コミュニティ名**と呼ばれるパスフレーズである。マネージャとエージェントのそれぞれにコミュニティ名をあらかじめ登録しておく。マネージャが送信する SNMP メッセージには必ずコミュニティ名が格納されているので，エージェントは正規のマネージャから送られたものかどうかを確認できる。その認証に成功したら，当該メッセージが実行される。

詳説

デフォルトのコミュニティ名は「public」になっている。認証の仕組みを有効に機能させるため，コミュニティ名を推測されにくいフレーズに変更するべきである

コミュニティ名による認証に失敗した場合，**認証トラップ**

(authenticationFailure) がエージェントからマネージャに対して通知することができる。この認証トラップの宛先アドレスは，あらかじめエージェントに登録しておく。

SNMPv1 では，コミュニティ名や MIB 情報を含め，パケットに格納されたデータは平文のまま送信される。

一方，SNMPv3 ではセキュリティが強化されており，認証とプライバシ（パケットデータの暗号化）の機能が追加された。認証機能では，従来のコミュニティ名以外の方式として，USM（User-based Security Model）が規定されている。これは，ユーザごとに，ユーザ名，パスワード，MIB へのアクセス権限をきめ細かく設定する仕組みである。パケットに格納されるパスワードは，タイムスタンプとハッシュを用いて暗号化されているため，盗聴やリプレイ攻撃を防止できる。プライバシ機能では，パケットデータの暗号化に共通鍵方式を用いる。なお，認証とプライバシの機能は，それぞれの使用／未使用を選択できる。

SNMPv3 対応のネットワーク機器を監視する際には，セキュリティモデルとして，従来のコミュニティ名と SNMPv3 の USM のどちらを用いるのか，マネージャとエージェントの両者で方式をそろえて設定する必要がある。

● MIB

監視の対象となるネットワーク機器の設定やステータスなどの情報は，本来，機器固有のデータ形式で管理されている。これを仮想的な管理オブジェクトとして表現し，機器間の差異をなくして統一的に取り扱えるよう，**MIB**（Management Information Base，管理情報ベース）が定義されている。ネットワーク機器は，MIB オブジェクトの値を静的に保存しているわけではない。エージェントは，マネージャから MIB オブジェクトに対する要求を受け取るたびに，関連するプログラムを内部で実行して，指定された MIB オブジェクトの値を取得したり，設定したりする。

MIB は，ASN.1（Abstract Syntax Notation One）と呼ばれる，ISO で定められた抽象構文記述法を用いて記述されており，様々なネットワーク機器のオブジェクトを表現することができる。

MIB は MIB オブジェクトのセット（集合）であり，次の図に

試験に出る

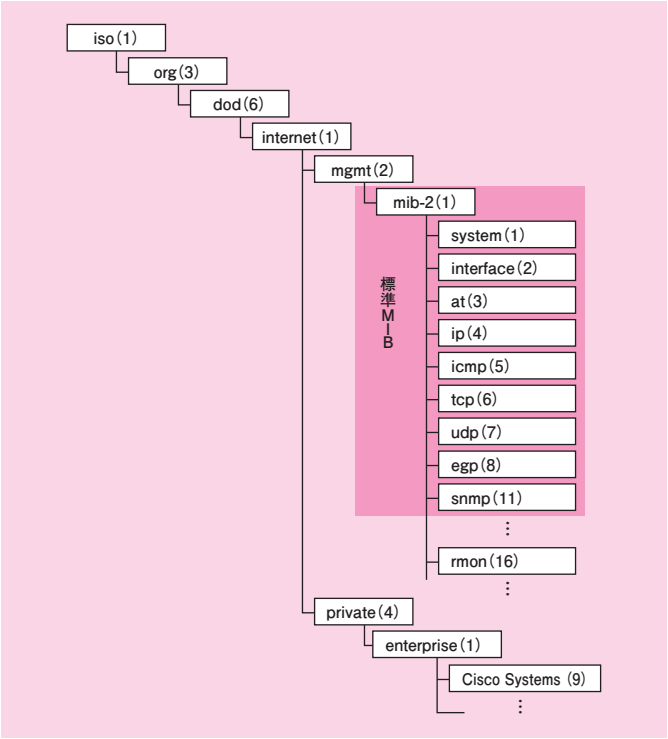
MIBについて，令和3年午後Ⅰ問1，令和元年午後Ⅰ問1，平成22年午後Ⅰ問2で出題された。RMONについて，令和4年午前Ⅱ問14で出題された

用語解説

RMON

RMON（Remote network MONitoring）とは，RMON MIB，RMON2 MIBによるネットワーク監視を行う技術である。RMON MIB，RMON2 MIBをサポートしている機器（L2SW，L3SWなど）は，同機器を通過するトラフィックについて，ポート毎の統計情報やプロトコル毎の統計情報などを蓄積している。監視サーバから同MIBの収集には，通常のMIBと同様，SNMPを用いる

示すようなツリー構造をしている。



図：MIB の構成

MIB で定義されるオブジェクトは，OID（Object IDentifier，オブジェクト識別子）と呼ばれる一意な名前をもち，次のように表現される。

```
system OBJECT IDENTIFIER ::= {
    iso org dod internet mgmt mib-2 system}
```

又は，次のようにピリオドで区切られた数字の列で表現される。

```
system OBJECT IDENTIFIER ::= .1.3.6.1.2.1.1
```

さらに，次のように上位の OID を用いて表現することも可能である。

```
system OBJECT IDENTIFIER ::= {mib-2 1}
```

MIB オブジェクトは一つ以上のインスタンスをもつ。ルータを例に挙げると，一般に複数のインタフェースをもつ。インタフェースに関する MIB オブジェクト（mib-2 interfaces の配下にあるもの）は，インタフェースごとにインスタンスが生成され，1 から始まるインスタンス番号によって識別される。MIB 変数を取得するときは，「x.y」という書式に従い，OID（x），ピリオド，インスタンス番号（y）の順番で指定する。例えば，「ifInOctets」（.1.3.6.1.2.1.2.2.10）は，インタフェースが受信したオクテット数を保持している MIB オブジェクトである。インスタンス番号が「1」であるインタフェースの「ifInOctets」を取得するには，「ifInOctets.1」とする。

一方，機器の構成情報に関する MIB オブジェクト（例えば，mib-2 system の配下にあるもの）は，インスタンスが一つしかない。このとき，インスタンス番号には「0」を指定する。例えば，「sysUpTime」（.1.3.6.1.2.1.1.3）は，起動してからの経過時間（100 分の 1 秒単位）を保持している MIB オブジェクトである。これを取得するには，「sysUpTime.0」とする。

標準 MIB（mib-2）は RFC 1213 で規定され，ネットワーク管理用の標準的なオブジェクトツリーとして定義されている。OID は以下のとおりである。

```
1.3.6.1.2.1：iso(1).org(3).dod(6).internet(1).
             mgmt(2).mib2(1)
```

MIB には標準以外にも，ベンダが独自に拡張した MIB がある。これを「**プライベート MIB**」あるいは「**拡張 MIB**」などと呼ぶ。OID は以下のとおりである。

```
1.3.6.1.4.1：iso(1).org(3).dod(6).internet(1).
             private(4).enterprise(1)
```

この直下に，ベンダごとに private enterprise number が割り当てられており，それ以下のサブツリーでは，各社が独自に MIB オブジェクトを定義している。例えば，Cisco Systems 社の private enterprise number は「9」である（1.3.6.1.4.1.9）。通常，プライベート MIB の内容は標準 MIB より数が多いので，ネットワーク監視の対象となる。

ifInOctets と ifOutOctets のデータ型は，32 ビットのカウンタ値である。RFC1152 は，32 ビットのカウンタ値を次のように規定している。「（カウンタ値は）非負の整数を表し，最大値に達するまで単調に増加し続ける。その後，カウンタはラップしてゼロから再度開始される。この規約では，カウンタの最大値を $2^{32}-1$（10 進数で 4,294,967,295）と定義する」。つまり，カウンタ値が 4,294,967,295 である場合，1 が加算されると桁あふれを起こしてしまい，カウンタ値が 0 になる。このように，最大値を超えたときに 0 に戻ってから増加する動作をラップ（wrap）という。64 ビットのカウンタ（ハイカウンタ）である ifHCInOctets，ifHCOutOctets も MIB に定義されている。ネットワークの広帯域化に伴って用いられるようになってきた

MIB 情報の ifInOctets と ifOutOctets のラップについて，令和 3 年午後Ⅱ問 2，平成 22 年午後Ⅰ問 2 で出題された。64 ビットのカウンタ（ハイカウンタ）について，令和 3 年午後Ⅱ問 2 の本文中で言及された（出題はされていない）