Chương 4

Các phiên bản SNMP

- SNMPv1 : cấu trúc bản tin, cấu trúc MIB
- SNMPv2c : phương thức GetBulk và Notification, cấu trúc bản tin, cấu trúc MIB
- SNMPv3 : giải thuật authentication, privacy, trình tự khai báo snmpv3.

1. Các phiên bản SNMP

Các phiên bản SNMP khác nhau những gì ?

+ Khác nhau ở phương thức hoạt động (operation) : SNMPv1 có 5 phương thức, tuy nhiên các version khác sau này được bổ sung thêm một số phương thức mới.

+ Khác nhau ở cấu trúc bản tin SNMP (message format) : các phiên bản khác nhau sẽ khác nhau ở cấu trúc các bản tin.

Có những phiên bản SNMP nào ?

+ SNMPv1 : phiên bản đầu tiên của SNMP, có 5 phương thức Get, GetNext, Set, Response, Trap.

+ SNMPv2c : SNMP version 2 chia thành 2 phiên bản khác nhau ở cơ chế bảo mật, trong đó phiên bản vẫn sử dụng cơ chế bảo mật dựa vào community string như ở SNMPv1 gọi là Community-based SNMPv2 hay SNMPv2c. Một số tài liệu đã ghi chú không đúng rằng "SNMPv2c bổ sung thêm cơ chế community string so với SNMPv1", thực sự SNMPv2c và SNMPv1 đều có cơ chế xác thực đơn giản bằng community giống nhau.

+ SNMPv2u : đây là phiên bản SNMPv2 sử dụng cơ chế bảo mật có chứng thực bằng băm[1] và mã hóa đối xứng[2] dữ liệu, gọi là User-based SNMPv2 hay SNMPv2u. Sau này phiên bản SNMPv3 ra đời đã thay thế hoàn toàn SNMPv2u và người ta không còn ưu tiên dùng SNMPv2u nữa. Do đó SNMPv2u sẽ không được trình bày trong tài liệu này mà SNMPv3 sẽ được trình bày chi tiết. Trong thực tế rất khó tìm thấy một thiết bị còn hỗ trợ SNMPv2u.

+ SNMPv3 : phiên bản bảo mật nhất của SNMP sử dụng mô hình bảo mật dựa trên người dùng (User-based security model) với các cơ chế chứng thực bằng băm (MD5, SHA) và mã hóa (DES, AES) hiện đại. Việc lập trình ứng dụng hỗ trợ được SNMPv3 phức tạp hơn, do đó hầu hết các phần mềm SNMP manager phiên bản có hỗ trợ SNMPv3 đều có tính phí, trong khi phiên bản miễn phí chỉ hỗ trợ SNMPv1 và SNMPv2.

Tại sao cần phải biết sự khác nhau ở các phiên bản ?

Nếu công việc của bạn chỉ là ứng dụng được một phần mềm SNMP để quản lý các thiết bị trong công ty thì bạn chỉ cần biết 2 việc : thiết bị nào của bạn hỗ trợ các version SNMP nào; và phần mềm SNMP manager mà bạn sở hữu có hỗ trợ version SNMP tương ứng hay không. Nếu vậy thì bạn có thể dừng đọc quyển tài liệu này ở đây vì các phần sau này là không thích hợp. Hầu hết các tài liệu về SNMP đều không trình bày kỹ các phiên bản khác nhau vì hầu hết người đọc không cần để ý đến chúng.

Nếu bạn là chuyên viên bậc cao cần có kỹ năng giải quyết ở mức debug các vấn đề liên quan đến tương thích version của SNMP, chẳng hạn một phần mềm nào đó không thể quản lý một thiết bị của bạn, thì bạn cần tìm hiểu sự khác nhau giữa các version.

Và nếu bạn là người lập trình ứng dụng SNMP thì việc hiểu rõ các version khác nhau là yêu cầu bắt buộc, phần mềm của bạn cần có khả năng tương thích các thiết bị hỗ trợ version khác nhau.

Các tài liệu liên quan đến các phiên bản SNMP

Việc tìm hiểu các phiên bản SNMP tốn nhiều thời gian vì có nhiều đặc tả RFC liên quan đến chúng. Trong khuôn khổ quyển tài liệu này tác giả không thể trình bày hết các vấn đề ngoài các đặc điểm chính. Bảng sau liệt kê các RFC chủ yếu của các phiên bản SNMP :

RFC	Năm công bố	Nội dung	Thay thế
RFC1155 - Structure and Identification of Management Information for TCP/IP-based Internets	1990	Cấu trúc mib của các thiết bị chạy trên nền TCP/IP (SMIv1)	RFC1065
RFC1156 - Management Information Base for Network Management of TCP/IP-based internets	1990	Mib chuẩn của internet version 1 (Internet-standard mib), còn gọi là mib-1	RFC1066

[1] Băm (hash) là phương pháp mã hóa một văn bản nguồn (message) thành một chuỗi (digest) ngắn hơn nhiều lần và không thể giải ngược từ digest thành message, hash còn được gọi là mã hóa 1 chiều. Các phương pháp hash phổ biến hiện nay là MD5 và SHA.

[2] Mã hóa đối xứng (symmetric encryption) là phương pháp mã hóa dùng cùng 1 khóa để mã hóa và giải mã, khác với mã hóa bất đối xứng là dùng khóa mã hóa và khóa giải mã khác nhau.

RFC1157 - A Simple Network Management Protocol (SNMP)	1990	Đặc tả giao thức SNMPv1	RFC1098
RFC1213 - Management Information Base for Network Management of TCP/IP-based internets: MIB-II	1991	Mib chuẩn của internet phiên bản 2, còn gọi là mib-2	RFC1158
RFC2790 - Host Resources MIB	2000	Cấu trúc MIB của thiết bị dạng host (server)	RFC1514
RFC1901 - Introduction to Community-based SNMPv2	1996	Tài liệu ngắn gọn đầu tiên giới thiệu SNMPv2	
RFC2578 - Structure of Management Information Version 2 (SMIv2)	1999	Cấu trúc mib của các thiết bị chạy trên nền TCP/IP, phiên bản 2 (SMIv2)	RFC1902
RFC2579 - Textual Conventions for SMIv2	1999	Định nghĩa các kiểu dữ liệu dạng text của SMIv2	RFC1903
RFC3416 - Version 2 of the Protocol Operations for the Simple Network Management Protocol	2002	Đặc tả các phương thức hoạt động của SNMPv2	RFC1905
RFC3418 - Management Information Base for the Simple Network Management Protocol	2002	Cấu trúc MIB của SNMPv2	RFC1907
RFC1910 - User-based Security Model for SNMPv2	1996	Đặc tả mô hình bảo mật của phiên bản SNMPv2u	
RFC3412 - Message Processing and Dispatching for the Simple Network Management Protocol	2002	Mô tả cấu trúc bản tin SNMPv3	RFC2572
RFC3414 - User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)	2002	Mô hình bảo mật của phiên bản SNMPv3	RFC2574

Cột "Thay thế" là các RFC cũ của cùng nội dung trước đó, người đọc cần chú ý cập nhật RFC mới nhất. Có một số tài liệu SNMP được biên soạn trước khi các RFC mới ra đời nên nó dẫn giải các RFC đã lỗi thời (obsolete). Chẳng hạn về SNMPv2 trước đây có các RFC từ 1901 đến 1908, tuy nhiên hiện tại các RFC1902, 1903, 1904, 1905, 1906, 1907 đã được thay thế bằng các RFC2578, 2579, 2580, 3416, 3417, 3418.

Các RFC có thể được tìm tại 2 nguồn sau : http://tools.ietf.org/html/ hoặc http://www.faqs.org/rfcs/. Để tra cứu một RFC nào đó có bị thay thế bởi một RFC khác mới hơn hay không, bạn hãy tìm tại http://www.faqs.org/rfcs/rfc-obsolete.html. Tại thời điểm bạn đọc quyển tài liệu này, có thể một số RFC được trích dẫn ở đây đã trở nên lỗi thời.

2. SNMPv1

Chương 1 đã trình bày các vấn đề liên quan đến SNMPv1 gồm : 5 phương thức hoạt động, cấu trúc bản tin; chương này sẽ trình bày ngắn gọn lại và thêm phần cấu trúc các PDU [3].

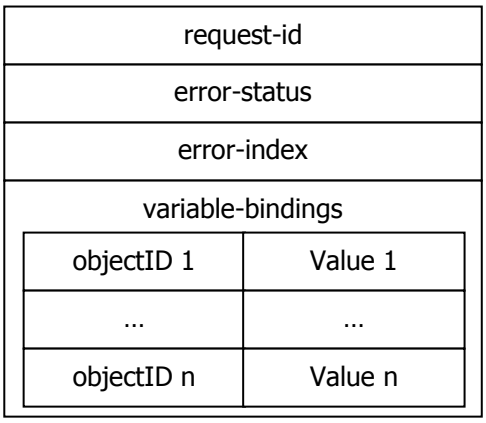
Các phương thức của SNMPv1

+ GetRequest : lấy thông tin của object có OID trong bản tin.
+ GetNextRequest : lấy thông tin của object nằm kế tiếp object có OID trong bản tin.
+ SetRequest : thiết lập giá trị cho object có OID trong bản tin.
+ GetResponse : trả về thông tin kết quả sau khi Get hoặc Set.
+ Trap : thông báo có sự kiện xảy ra tại agent.

Agent lắng nghe request ở cổng UDP 161 còn manager nhận trap ở cổng UDP 162.

[3] Cấu trúc của bản tin và của các PDU SNMPv1 được mô tả đầy đủ trong RFC1157

Cấu trúc của PDU GetRequest



Cấu trúc Get/GetNext/Set/Response PDU

+ request-id : mã số của request. ID này là số ngẫu nhiên do manager tạo ra, agent khi gửi bản tin GetResponse cho request nào thì nó phải gửi requestID giống như lúc nhận. Giữa manager và agent có thể có nhiều request & reponse, một request và một response là cùng một phiên trao đổi khi chúng có requestID giống nhau.

+ error-status : nếu = 0 là thực hiện thành công không có lỗi, nếu <> 0 là có lỗi xảy ra và giá trị của nó mô tả mã lỗi. Trong bản tin GetRequest, GetNextRequest, SetRequest thì error-status luôn = 0.

+ error-index : số thứ tự của objectid liên quan đến lỗi nếu có. Trong variable-bindings có nhiều objectid, được đánh số từ 1 đến n, một bản tin GetRequest có thể lấy cùng lúc nhiều object.

+ variable-bindings : danh sách các cặp [ObjectID – Value] cần lấy thông tin, trong đó objectId là định danh của object cần lấy, còn value không mang giá trị. Khi agent gửi bản tin trả lời thì nó sẽ copy lại bản tin này và điền vào value bằng giá trị của object.

Dùng một phần mềm bắt gói tin như Wireshark[4] bạn sẽ thấy cấu trúc của một bản tin GetRequest.

```
⊞ Frame 38 (85 bytes on wire, 85 bytes captured)
⊞ Ethernet II, Src: HewlettP_4d:dd:8c (00:1f:29:4d:dd:8c), Dst: SentecE&_71:0b:40 (00:0d:2
⊞ Internet Protocol, Src: 192.168.47.18 (192.168.47.18), Dst: 192.168.47.253 (192.168.47.2
⊞ User Datagram Protocol, Src Port: 4720 (4720), Dst Port: snmp (161)
⊟ Simple Network Management Protocol
    version: version-1 (0)
    community: public
  ⊟ data: get-request (0)
    ⊟ get-request
        request-id: 2142061952
        error-status: noError (0)
        error-index: 0
      ⊟ variable-bindings: 1 item
        ⊟ SNMPv2-MIB::sysUpTime.0 (1.3.6.1.2.1.1.3.0): unSpecified
          ⊟ Object Name: 1.3.6.1.2.1.1.3.0 (SNMPv2-MIB::sysUpTime.0)
              Scalar Instance Index: 0
            unSpecified
```

Trong hình trên là cấu trúc một bản tin SNMP với PDU là GetRequest. Bao gồm các thông tin :

+ version là v1, số 0 trong ngoặc là giá trị của trường version, nếu giá trị này là 0 nghĩa là version1.

+ community là "public".

+ request-id = 2142061952.

+ error-status = 0, nghĩa là không có lỗi. Trong bản tin GetResponse thì error-status mới được dùng.

+ error-index = 0.

+ phần variable-bindings bao gồm 1 item, mỗi item là 1 cặp objectid-value.

+ objectid là .1.3.6.1.2.1.1.3.0, theo mib-2 thì đó là sysUpTime.0

+ Scalar instance index = 0, đây là chỉ số index của sysUptime. Do một thiết bị chỉ có một khái niệm sysUptime nên index là 0 (sysUptime.0). Nếu bạn request ifDescr chẳng hạn thì mỗi interface sẽ có một description khác nhau và sẽ có index khác nhau.

+ value = unSpecified. Do bản tin là GetRequest nên value sẽ không mang giá trị, giá trị sẽ được ghi vào và trả về trong bản tin GetResponse.

[4] Wireshark là công cụ phân tích gói tin miễn phí, download tại http://wireshark.org

Cấu trúc của PDU GetResponse

+ request-id : mã số của request. ID này phải giống với request-id của bản tin GetRequest trước đó.

+ error-status : mang một trong các giá trị noError(0), tooBig(1), noSuchName(2), badValue(3), readOnly(4), genErr(5). Nếu agent lấy thông tin để trả lời request thành công thì error-status là noError(0).

+ objectid : định danh của object được trả về. Nếu trước đó là GetRequest thì objectid sẽ giống với objectid trong bản tin request, nếu trước đó là GetNextRequest thì objectid sẽ là định danh của object nằm sau (nằm sau trong mib) objectid của request.

Hình sau là bản tin trả lời cho GetRequest sysUpTime ở trên, với giá trị trả về là 109852988 (centi giây)

```
Simple Network Management Protocol
    version: version-1 (0)
    community: public
  data: get-response (2)
    get-response
        request-id: 2142061952
        error-status: noError (0)
        error-index: 0
      variable-bindings: 1 item
        SNMPv2-MIB::sysUpTime.0 (1.3.6.1.2.1.1.3.0): 109852988
          Object Name: 1.3.6.1.2.1.1.3.0 (SNMPv2-MIB::sysUpTime.0)
              Scalar Instance Index: 0
            SNMPv2-MIB::sysUpTime: 109852988
```

Cấu trúc của PDU GetNextRequest

Cấu trúc GetNextRequest giống với GetRequest, chỉ khác ở byte chỉ ra bản tin là GetNextRequest PDU.

Hình sau là bản tin GetNextRequest với objectid là sysContact, sau đó agent sẽ gửi bản tin GetReponse trả lời với objectid là sysName, vì sysName nằm sau sysContact trong mib. Chú ý request-id là giống nhau.

```
Simple Network Management Protocol
    version: version-1 (0)
    community: public
  data: get-next-request (1)
    get-next-request
        request-id: 2142061955
        error-status: noError (0)
        error-index: 0
      variable-bindings: 1 item
        SNMPv2-MIB::sysContact.0 (1.3.6.1.2.1.1.4.0): unSpecified
          Object Name: 1.3.6.1.2.1.1.4.0 (SNMPv2-MIB::sysContact.0)
              Scalar Instance Index: 0
            unSpecified
```

```
Simple Network Management Protocol
    version: version-1 (0)
    community: public
  data: get-response (2)
    get-response
        request-id: 2142061955
        error-status: noError (0)
        error-index: 0
      variable-bindings: 1 item
        SNMPv2-MIB::sysName.0 (1.3.6.1.2.1.1.5.0): C2950
          Object Name: 1.3.6.1.2.1.1.5.0 (SNMPv2-MIB::sysName.0)
              Scalar Instance Index: 0
            SNMPv2-MIB::sysName: C2950
```

Cấu trúc của PDU SetRequest

Cấu trúc SetRequest cũng giống với GetRequest, objectid-value chỉ ra đối tượng và giá trị cần set.

Hình sau là bản tin SetRequest đặt lại tên của thiết bị là "Cisco2950", tiếp theo agent sẽ gửi bản tin GetResponse thông báo gái trị của sysName sau khi set.

```
Simple Network Management Protocol
    version: version-1 (0)
    community: private
  data: set-request (3)
    set-request
        request-id: 2142061958
        error-status: noError (0)
        error-index: 0
      variable-bindings: 1 item
        SNMPv2-MIB::sysName.0 (1.3.6.1.2.1.1.5.0): Cisco2950
          Object Name: 1.3.6.1.2.1.1.5.0 (SNMPv2-MIB::sysName.0)
              Scalar Instance Index: 0
            SNMPv2-MIB::sysName: Cisco2950
```

```
Simple Network Management Protocol
    version: version-1 (0)
    community: private
  data: get-response (2)
    get-response
        request-id: 2142061958
        error-status: noError (0)
        error-index: 0
      variable-bindings: 1 item
        SNMPv2-MIB::sysName.0 (1.3.6.1.2.1.1.5.0): Cisco2950
          Object Name: 1.3.6.1.2.1.1.5.0 (SNMPv2-MIB::sysName.0)
              Scalar Instance Index: 0
            SNMPv2-MIB::sysName: Cisco2950
```

Cấu trúc của PDU Trap

Cấu trúc của bản tin trap của SNMPv1 như sau :

+ enterprise : kiểu của object gửi trap. Đây là một OID giúp nhận dạng thiết bị gửi trap là thiết bị gì; nhận dạng chi tiết đến hãng sản xuất, chủng loại, model. OID này bao gồm một chỉ số doanh nghiệp (enterprise number) và chỉ số id của thiết bị của hãng do hãng tự định nghĩa.

+ agent address : địa chỉ IP của nguồn sinh ra trap. Có thể bạn sẽ thắc mắc tại sao lại có IP của nguồn sinh ra trap trong khi bản tin IP chứa gói SNMP đã có địa chỉ nguồn. Giả sử mô hình giám sát của bạn như sau : tất cả trap sender được cấu hình để gửi trap đến một trap receiver trung gian, gọi là trap relay, sau đó trap relay mới gửi đến nhiều trap receiver cùng lúc; thì lúc này bản tin trap nhận được tại trap receiver sẽ có IP source là của trap relay, trong khi IP của nguồn phát sinh trap thực sự nằm trong agent address.

+ generic-trap : kiểu của các loại trap generic.

+ specific-trap : kiểu của các loại trap do người dùng tự định nghĩa.

+ time-stamp : thời gian tính từ lúc thiết bị được khởi động đến lúc gửi bản tin trap, tính bằng centi giây.

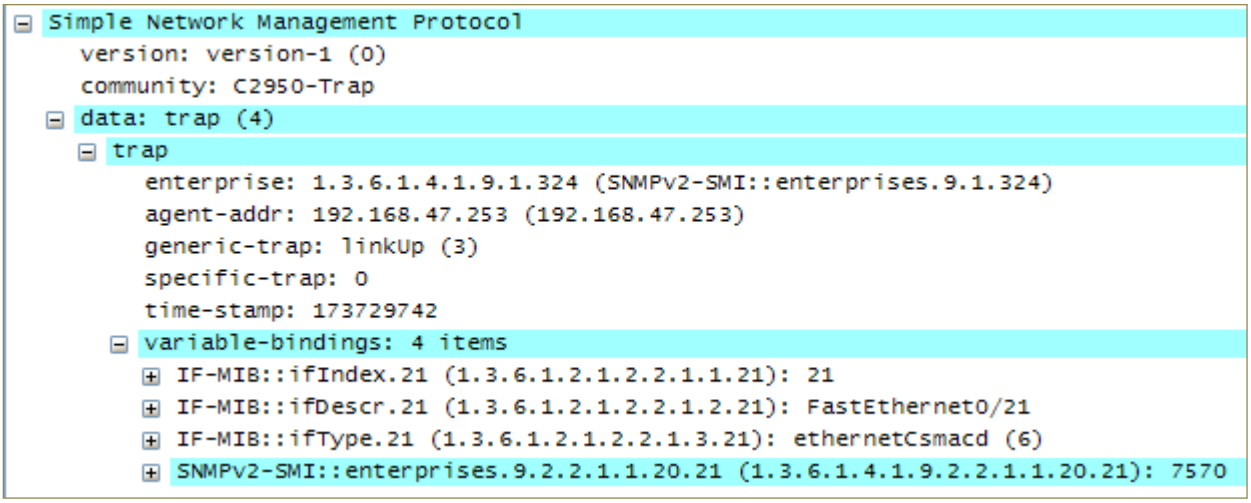
enterprise
agent-addr
generic-trap
specific-trap
time-stamp
variable-bindings

objectID 1	Value 1
…	…
objectID n	Value n

Cấu trúc Trap PDU

+ variable-bindings : các cặp objectID – value mô tả các object có liên quan đến trap.

Hình sau là bản tin trap thông báo interface FastEthernet0/21 đã UP.



+ enterprise = .1.3.6.1.4.1.9.1.324, đây là định danh của thiết bị Cisco switch Catalyst 2950 (.9.1.324)

+ agent-addr = 192.168.47.253

+ generic-trap = 3, cho biết đây là bản tin trap kiểu generic, giá trị 3 nghĩa là linkUp.

+ specific-trap = 0, do đây là trap kiểu generic nên không sử dụng đến specific.

+ time-stamp = 173729742.

+ variable-bindings gồm 4 item, chỉ ra 4 cặp objectid-value, gồm : ifIndex=21, ifDescr="FastEthernet0/21", ifType=6, và một object riêng của Cisco có value = 7570 (2 ký tự hexa 0x75 0x70 là chữ "up").

3. SNMPv2c

Khác biệt của SNMPv2c so với SNMPv1 là :

+ Có nhiều phương thức hơn so với SNMPv1.

+ Cấu trúc bản tin Trap PDU khác so với SNMPv1.

+ Có thêm bản tin Bulk PDU với cấu trúc riêng.

Các phương thức của SNMPv2c

SNMPv2c có 8 phương thức gồm : GetRequest, GetNextRequest, Response, SetRequest, GetBulkRequest, InformRequest, Trap và Report. Như vậy so với SNMPv1 thì v2c có thêm các phương thức GetBulk, Inform và Report.

+ GetRequest : manager gửi GetRequest cho agent để lấy thông tin.

+ GetNextRequest : manager gửi GetNextRequest cho agent để lấy thông tin của object nằm sau object được chỉ ra trong bản tin GetNext.

+ SetRequest : manager gửi SetRequest cho agent để thiết lập giá trị cho một object nào đó.

+ GetBulkRequest : phương thức này dùng để lấy một loạt nhiều object chỉ trong 1 bản tin GetBulk. Các bản tin Get/GetNext vẫn có thể lấy cùng lúc nhiều object bằng cách đưa tất cả chúng vào danh sách variable-bindings trong bản tin request, nhưng GetBulk có thể lấy nhiều object mà chỉ cần chỉ ra 1 object trong variable-bindings.

+ Response : agent gửi Response cho manager để thông báo kết quả của request mà nó nhận trước đó, Response là bản tin trả lời cho các Get/GetNext/GetBulk/Set/Inform request.

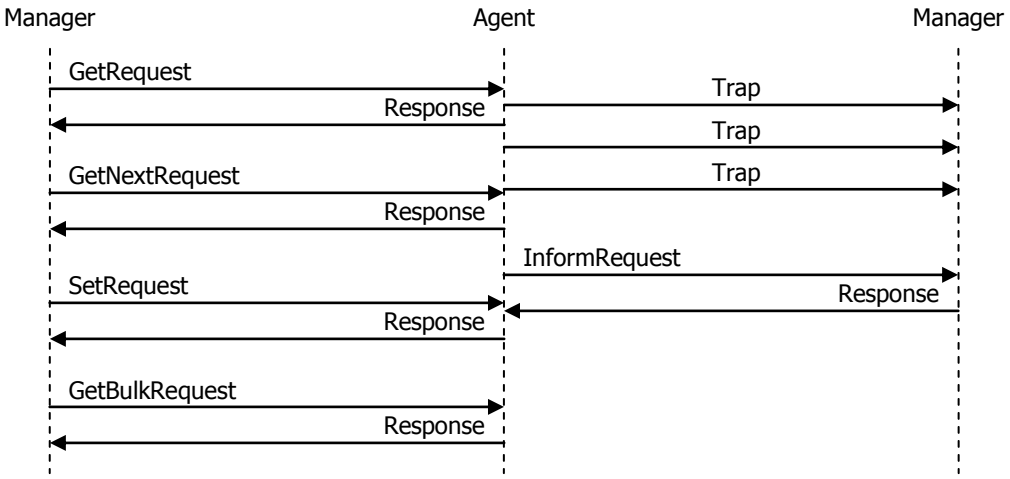
+ Trap : agent gửi Trap cho manager để thông báo về một sự kiện đang xảy ra tại agent.

+ InformRequest : có tác dụng tương tự như trap, nhưng khi manager nhận được InformRequest thì nó sẽ gửi lại Response để xác nhận đã nhận được thông báo, còn Trap thì không có cơ chế xác nhận.

+ Report : bản tin Report không được định nghĩa trong RFC3416, các hệ thống có sử dụng Report phải tự định nghĩa chúng, tuy nhiên bản tin Report vẫn có cấu trúc giống như các bản tin khác.

Agent lắng nghe request ở cổng UDP 161 còn manager nhận trap & inform ở cổng UDP 162.

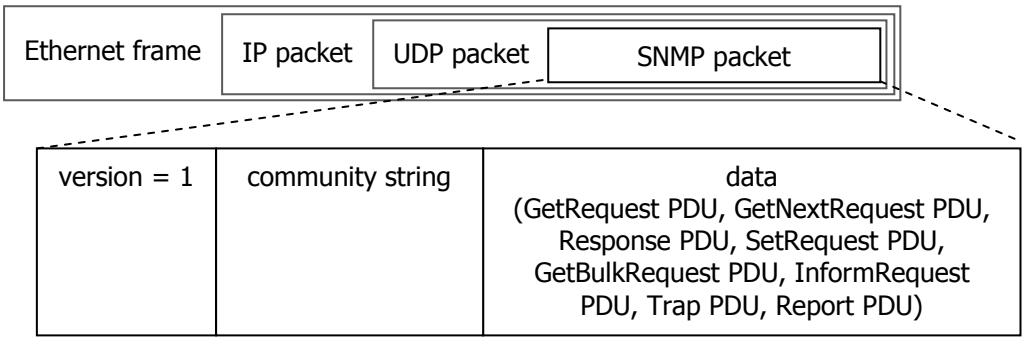
Hình sau minh họa hoạt động của các phương thức SNMPv2c :



Hình minh họa các phương thức của SNMPv2c

Cấu trúc bản tin SNMPv2c

Cấu trúc chung của bản tin SNMPv2c như sau [5]:



+ version : phiên bản SNMP (v1 = 0, v2c = 1, v2u = 2, v3 = 3).

+ community string : chuỗi community.

+ data : phần data là các bản tin ứng với các phương thức của SNMP.

Trong SNMPv2c, bản tin PDU có 2 loại cấu trúc là PDU và BulkPDU. Các bản tin GetRequest, GetNextRequest, SetRequest, Response, Trap, InformRequest và Report có cùng cấu trúc là PDU; còn GetBulkRequest có cấu trúc là BulkPDU [6].

Cấu trúc PDU

Cấu trúc PDU của SNMPv2c không thay đổi gì so với PDU của SNMPv1, gồm các trường :

+ request-id : mã số của request. ID này là số ngẫu nhiên do manager tạo ra, agent khi gửi bản tin Response cho request nào thì nó phải gửi requestID giống như lúc nhận. Giữa manager và agent có thể có nhiều request & reponse, một request và một response là cùng một phiên trao đổi khi chúng có requestID giống nhau.

+ error-status : nếu = 0 là thực hiện thành công không có lỗi, nếu <> 0 là có lỗi xảy ra và giá trị của nó mô tả mã lỗi. Trong các bản tin request thì error-status luôn = 0.

+ error-index : số thứ tự của objectid liên quan đến lỗi nếu có. Trong variable-bindings có nhiều objectid, được đánh số từ 1 đến n.

[5] Cấu trúc của bản tin SNMPv2 được mô tả trong RFC1901, trang 5

[6] Cấu trúc của các PDU SNMPv2c được mô tả trong RFC3416

+ variable-bindings : danh sách các cặp [ObjectID – Value] cần lấy thông tin, trong đó objectId là định danh của object cần lấy, còn value là giá trị của object đó. Khi agent gửi bản tin request thì value là không xác định, khi gửi trả lời thì nó sẽ điền vào value bằng giá trị của object.

request-id	
error-status	
error-index	
variable-bindings	
objectID 1	value 1
…	…
objectID n	value n

Cấu trúc Get/GetNext/Set/Response/Trap/Inform PDU

Cấu trúc Bulk PDU

GetBulkRequest có thể lấy về nhiều object mà chỉ cần chỉ ra một vài object trong bản tin gửi đi. Nguyên lý của nó là khai báo số lượng object tính từ object được chỉ ra trong request mà agent phải lần lượt trả về thông tin, kiểu như "hãy lấy cho tôi 20 object tính từ object có id là ...". Một bản tin GetBulk bao gồm các trường :

+ request-id : tương tự như cấu trúc của PDU.

+ non-repeaters : số lượng item đầu tiên trong variable-bindings của GetBulk mà agent phải trả lời bằng item nằm kế tiếp trong mib, mỗi item trong request thì sẽ có một item trong response.

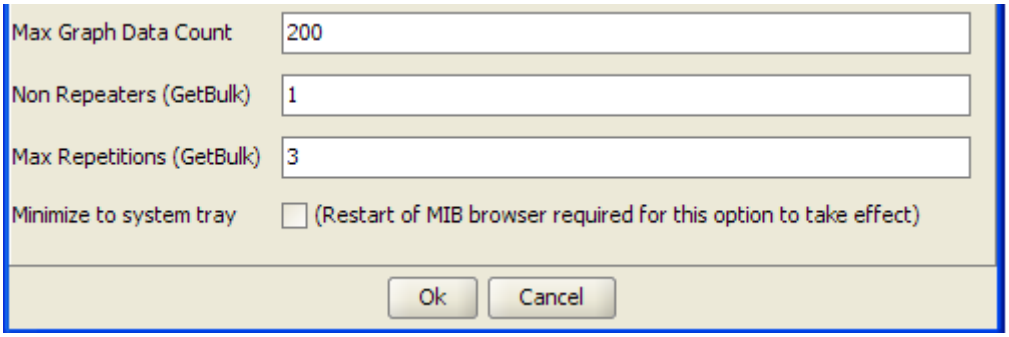
+ max-repetitions : các item còn lại trong variable-bindings sẽ được agent trả lời bằng *max-repetitions* item nằm kế tiếp chúng trong mib, mỗi item còn lại trong request này sẽ có *max-repetitions* item tương ứng trong response.

request-id	
non-repeaters	
max-repetitions	
variable-bindings	
objectID 1	value 1
…	…
objectID n	value n

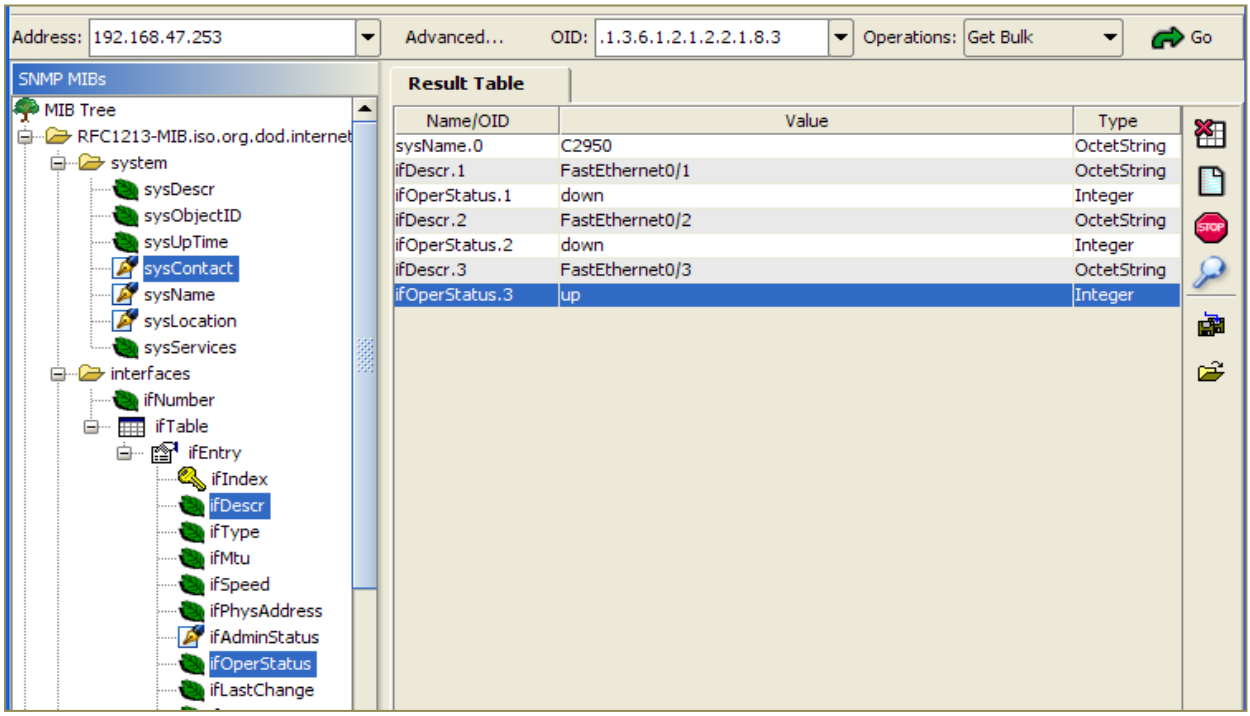
Cấu trúc GetBulk PDU

Ví dụ 1 : gửi bản tin GetBulkRequest để lấy tên của thiết bị, mô tả & tình trạng hoạt động của 3 interface đầu tiên, dùng iReasoning Mib Browser.

+ Trên iReasoning Mib Browser, vào menu Tools/Options; đặt Non Repeaters = 1, Max Repetitions = 3.



+ Trên cây Mib, nhấn nút Ctrl và chọn cùng lúc các object sysContact, ifDescr, ifOperStatus; chọn Operations = GetBulk và nhấn nút Go.



+ Phần mềm sẽ gửi bản tin có non-repeaters = 1, max-repetitions = 3, variable-bindings có 3 item là sysContact, ifDescr, ifOperStatus như hình sau :

```
Simple Network Management Protocol
    version: v2c (1)
    community: public
  data: getBulkRequest (5)
    getBulkRequest
        request-id: 761049685
        non-repeaters: 1
        max-repetitions: 3
      variable-bindings: 3 items
        SNMPv2-MIB::sysContact.0 (1.3.6.1.2.1.1.4.0): unSpecified
        IF-MIB::ifDescr (1.3.6.1.2.1.2.2.1.2): unSpecified
        IF-MIB::ifOperStatus (1.3.6.1.2.1.2.2.1.8): unSpecified
```

+ Agent sẽ trả lời bằng bản tin Response có danh sách variable-bindings gồm 1 item sysName.0 và 3 cặp ifDescr + ifOperStatus.

```
Simple Network Management Protocol
    version: v2c (1)
    community: public
  data: get-response (2)
    get-response
        request-id: 761049685
        error-status: noError (0)
        error-index: 0
      variable-bindings: 7 items
        SNMPv2-MIB::sysName.0 (1.3.6.1.2.1.1.5.0): C2950
        IF-MIB::ifDescr.1 (1.3.6.1.2.1.2.2.1.2.1): FastEthernet0/1
        IF-MIB::ifOperStatus.1 (1.3.6.1.2.1.2.2.1.8.1): down (2)
        IF-MIB::ifDescr.2 (1.3.6.1.2.1.2.2.1.2.2): FastEthernet0/2
        IF-MIB::ifOperStatus.2 (1.3.6.1.2.1.2.2.1.8.2): down (2)
        IF-MIB::ifDescr.3 (1.3.6.1.2.1.2.2.1.2.3): FastEthernet0/3
        IF-MIB::ifOperStatus.3 (1.3.6.1.2.1.2.2.1.8.3): up (1)
```

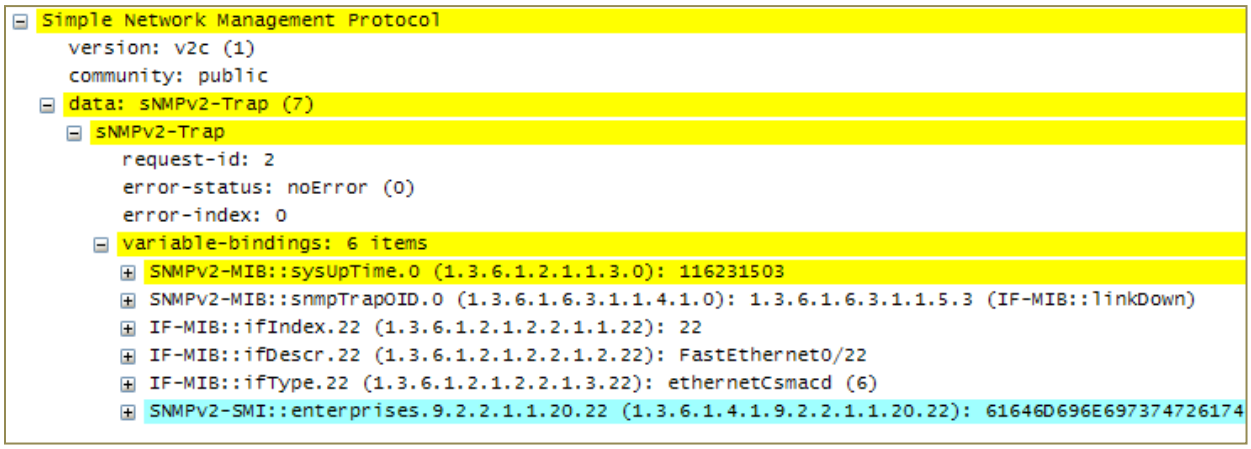
+ Do bản tin request có non-repeaters = 1 nên agent sẽ trả lời (không lặp lại) cho 1 item đầu tiên trong GetBulkRequest là sysContact. Vì nằm sau sysContact là sysName nên item response đầu tiên là sysName.0.

+ Do bản tin request có max-repetitions = 3 nên agent sẽ trả lời lặp lại 3 lần cho các item còn lại trong GetBulkRequest là ifDescr và ifOperStatus. Vì vậy các item còn lại trong response sẽ lần lượt là 3 cặp ifDescr & ifOperStatus.

SNMPv2 Trap PDU và InformRequest PDU

Bản tin Trap và Inform có cùng cấu trúc PDU như các bản tin khác. Trong SNMPv2, các bản tin này khi gửi đi thì 2 item đầu tiên trong variable-bindings phải là sysUpTime.0 và snmpTrapOID.0, sau đó mới đến các item liên quan đến sự kiện. Trong khi SNMPv1 Trap chỉ chứa các item liên quan đến sự kiện.

Hình sau minh họa một trap SNMPv2



Đọc sysUpTime.0 thì trap receiver biết được tại thời điểm mà agent phát ra trap thì agent đã hoạt động được bao lâu. Đọc snmpTrapOID.0 thì trap receiver có thể biết được ý nghĩa của bản tin trap là gì. Trong hình trên, snmpTrapOID.0 có giá trị .1.3.6.1.6.3.1.1.5.3, id này là của trap linkDown [7]. Tất nhiên phần mềm nhận trap (Wireshark) phải hiểu được TrapOID này nghĩa là gì thì mới hiển thị được chữ "IF-MIB::linkDown", nếu bạn dùng một phần mềm trap receiver không hiểu TrapOID này là gì thì nó chỉ hiển thị chuỗi id mà không có chú thích "linkDown". Chẳng hạn item cuối cùng trong bản tin trên là một trap của riêng Cisco nên phần mềm không thể có chú thích gì thêm.

Các item khác cho biết thêm thông tin về object đang bị down như index = 22, description = FastEthernet0/22.

4. SNMPv3

Tóm tắt

+ SNMP có các phiên bản v1, v2c, v2u, v3.

+ SNMPv1 có 5 phương thức GetRequest, GetNextRequest, SetRequest, GetResponse và Trap.

+ Bản tin SNMPv1 có 2 loại PDU và Trap-PDU.

+ SNMPv2 có 8 phương thức GetRequest, GetNextRequest, SetRequest, Response, GetBulkRequest, Trap, InformRequest và Report.

+ Bản tin SNMPv2 có 2 loại PDU và Bulk-PDU.

[7] Trap linkDown được định nghĩa trong RFC2863 – The Interfaces Group MIB, trang 48.