**MỞ ĐẦU**

Ngày nay, mạng máy tính đang trở nên phổ biến hơn rất nhiều. Những ứng dụng hiện nay được tập trung phát triển trên mạng Internet với một tốc độ chóng mặt cả về chất lượng và số lượng. Cộng đồng người sử dụng mạng và khối lượng tài nguyên, thông tin được chia sẻ thông qua môi trường mạng đang gia tăng từng ngày, từng giờ. Những tiện ích mà mạng máy tính mang lại cho chúng ta là không thể phủ nhận. Ở bất kì nơi nào, chúng ta cũng đều nhận thấy sự xuất hiện của mạng máy tính, với nhiều hình thức, tổ chức và quy mô lớn nhỏ khác nhau. Tất cả đều liên kết với nhau để tạo nên một thế giới ảo mạnh mẽ, rộng lớn và phát triển không ngừng.

Việc áp dụng các mô hình mạng máy tính ở các quy mô công ty, tổ chức, doanh nghiệp đang được triển khai rộng khắp, trong đó có Việt Nam. Điều này đòi hỏi nhiều yếu tố về nghiệp vụ, chuyên môn, yêu cầu đặc thù của các tổ chức. Từ đó, các nhà quản lí sẽ xây dựng các mô hình mạng phù hợp với các điều kiện đó. Chính vì thế, việc tổ chức, quản lí, bảo trì mạng máy tính là hết sức quan trọng. Nhận thấy được vấn đề đó, nhóm đã quyết định lựa chọn đề tài: Quản lí mạng máy tính thông qua công cụ Cacti trên nền Linux với giao thức SNMP. Đây là một bộ công cụ rất mạnh cho phép quản lí tài nguyên của một mạng với quy mô mở rộng, đảm bảo tính nhất quán và khoa học trong khâu tổ chức cũng như bảo trì.

Trong suốt quá trình thực hiện đề tài, nhóm luôn nỗ lực tìm hiểu nguyên lí, các thức triển khai công cụ cũng như các vấn đề liên quan tới mô hình mạng. Tuy nhiên, chắc chắn vẫn còn những thiếu xót không thể tránh khỏi và cần tiếp tục cải thiện trong tương lai. Nhóm xin chân thành cảm ơn sự hướng dẫn, giúp đỡ tận tình của TS. Bành Thị Quỳnh Mai để hoàn thành tốt đề tài này.

*Nhóm sinh viên*

1. Tổng quan về giao thức SNMP
2. Khái niệm cơ bản

SNMP là “giao thức quản lý mạng đơn giản”, dịch từ cụm từ “Simple Network Management Protocol”.

Giao thức là một tập hợp các thủ tục mà các bên tham gia cần tuân theo để có thể giao tiếp được với nhau. Trong lĩnh vực thông tin, một giao thức quy định cấu trúc, định dạng (format) của dòng dữ liệu trao đổi với nhau và quy định trình tự, thủ tục để trao đổi dòng dữ liệu đó. Nếu một bên tham gia gửi dữ liệu không đúng định dạng hoặc không theo trình tự thì các bên khác sẽ không hiểu hoặc từ chối trao đổi thông tin. SNMP là một giao thức, do đó nó có những quy định riêng mà các thành phần trong mạng phải tuân theo.

Cốt lõi của SNMP là một tập hợp đơn giản các hoạt động giúp nhà quản trị mạng có thể quản lý, thay đổi trạng thái của mạng. Ví dụ chúng ta có thể dùng SNMP để tắt một interface nào đó trên router của mình, theo dõi hoạt động của card Ethernet, hoặc kiểm soát nhiệt độ trên switch và cảnh báo khi nhiệt độ quá cao.

Một thiết bị hiểu được và hoạt động tuân theo giao thức SNMP được gọi là “có hỗ trợ SNMP” (SNMP supported) hoặc “tương thích SNMP” (SNMP compartible).

SNMP dùng để quản lý, nghĩa là có thể theo dõi, có thể lấy thông tin, có thể được thông báo, và có thể

tác động để hệ thống hoạt động như ý muốn. VD một số khả năng của phần mềm SNMP :

* Theo dõi tốc độ đường truyền của một router, biết được tổng số byte đã truyền/nhận.
* Lấy thông tin máy chủ đang có bao nhiêu ổ cứng, mỗi ổ cứng còn trống bao nhiêu.
* Tự động nhận cảnh báo khi switch có một port bị down.
* Điều khiển tắt (shutdown) các port trên switch.

SNMP dùng để quản lý mạng, nghĩa là nó được thiết kế để chạy trên nền TCP/IP và quản lý các thiết bị có nối mạng TCP/IP. Các thiết bị mạng không nhất thiết phải là máy tính mà có thể là switch, router, firewall, adsl gateway, và cả một số phần mềm cho phép quản trị bằng SNMP. Giả sử bạn có một cái máy giặt có thể nối mạng IP và nó hỗ trợ SNMP thì bạn có thể quản lý nó từ xa bằng SNMP.

SNMP là giao thức đơn giản, do nó được thiết kế đơn giản trong cấu trúc bản tin và thủ tục hoạt động, và còn đơn giản trong bảo mật (ngoại trừ SNMP version 3). Sử dụng phần mềm SNMP, người quản trị mạng có thể quản lý, giám sát tập trung từ xa toàn mạng của mình.

SNMP thường tích hợp vào trong router, nhưng khác với SGMP( Simple Gateway Management Protocol) được dùng chủ yếu cho các router Internet, SNMP có thể dùng để quản lý các hệ thống Unix, Window, máy in, nguồn điện… Nói chung, tất cả các thiết bị có thể chạy các phần mềm cho phép lấy được thông tin SNMP đều có thể quản lý được. Không chỉ các thiết bị vật lý mới quản lý được mà cả những phần mềm như web server, database.  
Một hướng khác của quản trị mạng là theo dõi hoạt động mạng, có nghĩa là theo dõi toàn bộ một mạng trái với theo dõi các router, host, hay các thiết bị riêng lẻ. RMON (Remote Network Monitoring) có thể giúp ta hiểu làm sao một mạng có thể tự hoạt động, làm sao các thiết bị riêng lẻ trong một mạng có thể hoạt động đồng bộ trong mạng đó IETF (Internet Engineering Task Force) là tổ chức đã đưa ra chuẩn SNMP thông qua các RFC.

* SNMP version 1 chuẩn của giao thức SNMP được định nghĩa trong RFC 1157 và là một chuẩn đầy đủ của IETF. Vấn đề bảo mật của SNMP v1 dựa trên nguyên tắc cộng đồng, không có nhiều password, chuổi văn bản thuần và cho phép bất kỳ một ứng dụng nào đó dựa trên SNMP có thể hiểu các hiểu các chuổi này để có thể truy cập vào các thiết bị quản lý. Có 3 tiêu chuẩn trong: read-only, read-write và trap.
* SNMP version 2: phiên bản này dựa trên các chuổi “community”. Do đó phiên bản này được gọi là SNMPv2c, được định nghĩa trong RFC 1905, 1906, 1907, và đây chỉ là bản thử nghiệm của IETF. Mạc dù chỉ là thử nghiệm nhưng nhiều nhà sản xuất đã đưa nó vào thực nghiệm.
* SNMP version 3: là phiên bản tiếp theo được IETF đưa ra bản đầy đủ. Nó được khuyến nghị làm bản chuẩn, được định nghĩa trong RFC 1905, RFC 1906, RFC 1907, RFC 2571, RFC 2572, RFC 2573, RFC 2574 và RFC 2575. Nó hổ trợ các loại truyền thông riêng tư và có xác nhận giữa các thực thể.

Trong SNMP có 3 vấn đề cần quan tâm: Manager, Agent và MIB (Management Information Base). MIB là cơ sở dữ liệu dùng phục vụ cho Manager và Agent.   
**Manager** là một server có chạy các chương trình có thể thực hiện một số chức năng quản lý mạng. Manager có thể xem như là NMS (Network Manager Stations). NMS có khả năng thăm dò và thu thập các cảnh báo từ các Agent trong mạng. Thăm dò trong việc quản lý mạng là “nghệ thuật” đặt ra các câu truy vấn đến các agent để có được một phần nào đó của thông tin. Các cảnh báo của agent là cách mà agent báo với NMS khi có sự cố xảy ra. Cảnh bảo của agent được gửi một cách không đồng bộ, không nằm trong việc trả lời truy vấn của NMS. NMS dựa trên các thông tin trả lời của agent để có các phương án giúp mạng hoạt động hiệu quả hơn. Ví dụ khi đường dây T1 kết nối tới Internet bị giảm băng thông nghiêm trọng, router sẽ gửi một thông tin cảnh báo tới NMS. NMS sẽ có một số hành động, ít nhất là lưu lại giúp ta có thể biết việc gì đã xảy ra. Các hành động này của NMS phải được cài đặt trước.   
**Agent** là một phần trong các chương trình chạy trên các thiết bị mạng cần quản lý. Nó có thể là một chương trình độc lập như các deamon trong Unix, hoặc được tich hợp vào hệ điều hành như IOS của Cisco trên router. Ngày nay, đa số các thiết bị hoạt động tới lớp IP được cài đặt SMNP agent. Các nhà sản xuất ngày càng muốn phát triển các agent trong các sản phẩm của họ công việc của người quản lý hệ thống hay quản trị mang đơn giản hơn. Các agent cung cấp thông tin cho NMS bằng cách lưu trữ các hoạt độn khác nhau của thiết bị. Một số thiết bị thường gửi một thông báo “tất cả đều bình thường” khi nó chuyển từ một trạng thái xấu sang một trạng thái tốt. Điều này giúp xác định khi nào một tình trạng có vấn đề được giải quyết.

Mối quan hệ giữa NMS và agent:



Không có sự hạn chế nào khi NMS gửi một câu truy vấn đồng thời agent gửi một cảnh báo.  
MIB có thể xem như là một cơ sở dữ liệu của các đối tượng quản lý mà agent lưu trữ được. Bất kỳ thông tin nào mà NMS có thẻ truy cập được đều được định nghĩa trong MIB. Một agent có thể có nhiều MIB nhưng tất cả các agent đều có một lọa MIB gọi là MIB-II được định nghĩa trong RFC 1213. MIB-I là bản gốc của MIB nhưng ít dùng khi MIB-II được đưa ra. Bất kỳ thiết bị nào hổ trợ SNMP đều phải hổ trợ MIB-II. MIB-II định nghĩa các tham số như tình trạng của interface (tốc độ của interface, MTU, các octet gửi, các octet nhận. ...) hoặc các tham số gắn liền với hệ thống (định vị hệ thống, thông tin liên lạc với hệ thống, ...). Mục đích chính của MIB-II là cung cấp các thông tin quản lý theo TCP/IP. Có nhiều kiểu MIB giúp quản lý cho các mục đích khác nhau:

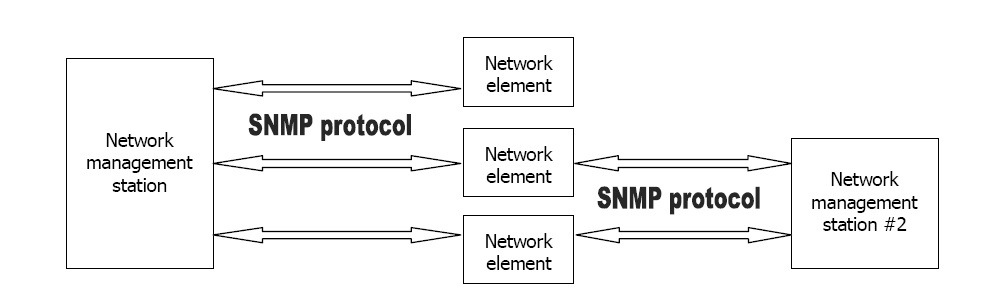
• ATM MIB (RFC 2515)   
• Frame Relay DTE Interface Type MIB (RFC 2115)   
• BGP Version 4 MIB (RFC 1657)   
• RDBMS MIB (RFC 1697)  
• RADIUS Authentication Server MIB (RFC 2619)  
• Mail Monitoring MIB (RFC 2249)  
• DNS Server MIB (RFC 1611)

Nhưng nhà sản xuất cũng như người dùng có thể định nghĩa các biến MIB riêng cho họ trong từng tình huống quản lý của họ.  
Quản lý Host Resource cũng là một phần quan trọng của quản lý mạng. Trước đây, sự khác nhau giữa quản lý hệ thống kiểu cũ và quản lý mạng không được xác định, nhưng bây giờ đã được phân biệt rõ ràng. RFC 2790 đưa ra Host Resource với định nghĩa tập hợp cá đối tượng cần quản lý trong hệ thống Unix và Window. Một số đối tượng đó là: dung lượng đĩa, số user của hệ thống, số tiến trình đang chạy của hệ thống và các phần mềm đã cài vào hệ thống. Trong một thế giới thương mại điện tử, các dịch vụ như web ngày càng trở nên phổ biến nên việc đảm bảo cho các server hoạt động tốt là việc hết sức quan trọng RMON (Remote Monitoring) hay còn gọi là RMON v1 được định nghĩa trong RFC 2819. RMON v1 cung cấp cho NMS các thông tin dạng packet về các thực thể trong LAN hay WAN. RMON v2 được xây dựng trên RMON v1 bởi những nhà cung cấp mạng và cung cấp thông tin ở lớp application. Thông tin có thể thu được bằng nhiều cách.. Một cách trong đó là đặt một bộ phận thăm dò của RMON trên mỗi phân đoạn mạng muốn theo dõi. RMON MIB được thiết kế để các RMON có thể chạy khi không kết nối logic giữa NMS và agent, có thể lấy được thông tin mà không cần chờ truy vấn của NMS. Sau đó, khi NMS muốn truy vấn, RMON sẽ trả lời bằng các thông tin thu thập được. Một đặc tính khác là ta có thể đặt ngưỡng cho một loại lỗi nào đó, và khi lỗi vượt quá ngưỡng đặt ra, RMON gửi một cảnh báo cho NMS.

1. Các thành phần chính

Theo RFC1157, kiến trúc của SNMP bao gồm 2 thành phần : các trạm quản lý mạng (network management station) và các thành tố mạng (network element).

Network management station thường là một máy tính chạy phần mềm quản lý SNMP (SNMP management application), dùng để giám sát và điều khiển tập trung các network element.



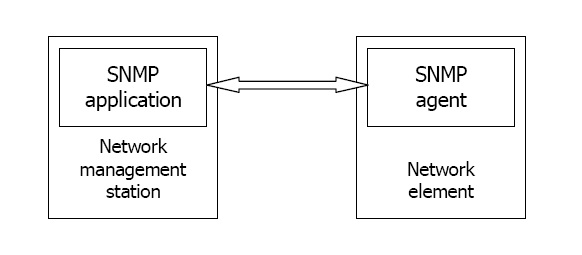
Network element là các thiết bị, máy tính, hoặc phần mềm tương thích SNMP và được quản lý bởi network management station. Như vậy element bao gồm device, host và application.

Một management station có thể quản lý nhiều element, một element cũng có thể được quản lý bởi nhiều management station. Vậy nếu một element được quản lý bởi 2 station thì điều gì sẽ xảy ra ? Nếu station lấy

thông tin từ element thì cả 2 station sẽ có thông tin giống nhau. Nếu 2 station tác động đến cùng một element thì element sẽ đáp ứng cả 2 tác động theo thứ tự cái nào đến trước.

Ngoài ra còn có khái niệm SNMP agent. SNMP agent là một tiến trình (process) chạy trên network element, có nhiệm vụ cung cấp thông tin của element cho station, nhờ đó station có thể quản lý được element. Chính xác hơn là application chạy trên station và agent chạy trên element mới là 2 tiến trình SNMP trực tiếp liên hệ với nhau. Các ví dụ minh họa sau đây sẽ làm rõ hơn các khái niệm này :

* Để dùng một máy chủ (station) quản lý các máy con (element) chạy HĐH Windows thông qua SNMP thì bạn phải : cài đặt một phần mềm quản lý SNMP (application) trên máy chủ, bật SNMP service (agent) trên máy con.
* Để dùng một máy chủ (station) giám sát lưu lượng của một router (element) thì bạn phải : cài phần mềm quản lý SNMP (application) trên máy chủ, bật tính năng SNMP (agent) trên router.



**Object ID**

Một thiết bị hỗ trợ SNMP có thể cung cấp nhiều thông tin khác nhau, mỗi thông tin đó gọi là một object. VD:

* Máy tính có thể cung cấp các thông tin : tổng số ổ cứng, tổng số port nối mạng, tổng số byte đã truyền/nhận, tên máy tính, tên các process đang chạy, ….
* Router có thể cung cấp các thông tin : tổng số card, tổng số port, tổng số byte đã truyền/nhận, tên router, tình trạng các port của router, …. Mỗi object có một tên gọi và một mã số để nhận dạng object đó, mã số gọi là Object ID (OID). VD :

+ Tên thiết bị được gọi là sysName, OID là 1.3.6.1.2.1.1.5

+ Tổng số port giao tiếp (interface) được gọi là ifNumber, OID là 1.3.6.1.2.1.2.1.

+ Địa chỉ Mac Address của một port được gọi là ifPhysAddress, OID là 1.3.6.1.2.1.2.2.1.6.

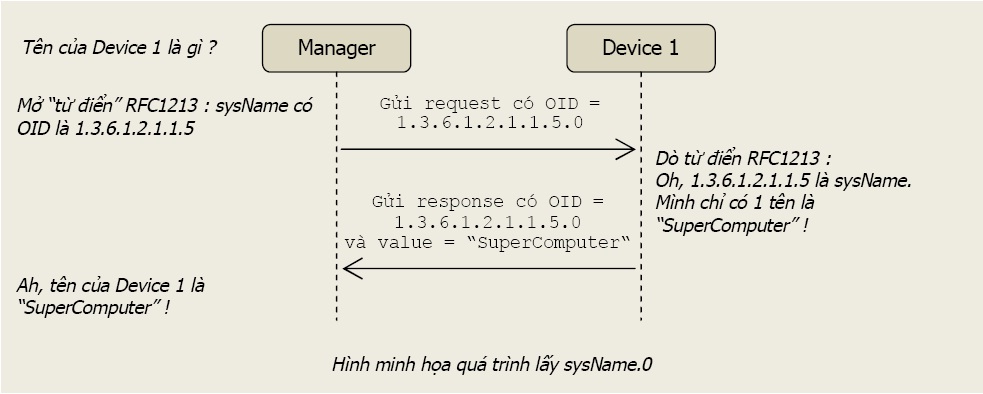
+ Số byte đã nhận trên một port được gọi là ifInOctets, OID là 1.3.6.1.2.1.2.2.1.10.

Ở hầu hết các thiết bị, các object có thể có nhiều giá trị thì thường được viết dưới dạng có sub-id. VD: một thiết bị dù chỉ có 1 tên thì nó vẫn phải có OID là sysName.0 hay 1.3.6.1.2.1.1.5.0. Ta cần nhớ quy tắc này để ứng dụng trong lập trình phần mềm SNMP manager.

Sub-id không nhất thiết phải liên tục hay bắt đầu từ 0. VD một thiết bị có 2 mac address thì có thể được gọi là ifPhysAddress.23 và ifPhysAddress. 125645.

OID của các object phổ biến có thể được chuẩn hóa, OID của các object do bạn tạo ra thì ta phải tự mô tả chúng. Để lấy một thông tin có OID đã chuẩn hóa thì SNMP application phải gửi một bản tin SNMP có chứa OID của object đó cho SNMP agent, SNMP agent khi nhận được thì nó phải trả lời bằng thông tin ứng với OID đó.

VD : Muốn lấy tên của một PC chạy Windows, tên của một PC chạy Linux hoặc tên của một router thì SNMP application chỉ cần gửi bản tin có chứa OID là 1.3.6.1.2.1.1.5.0. Khi SNMP agent chạy trên PC Windows, PC Linux hay router nhận được bản tin có chứa OID 1.3.6.1.2.1.1.5.0, agent lập tức hiểu rằng đây là bản tin hỏi sysName.0, và agent sẽ trả lời bằng tên của hệ thống. Nếu SNMP agent nhận được một OID mà nó không hiểu (không hỗ trợ) thì nó sẽ không trả lời.



Một trong các ưu điểm của SNMP là nó được thiết kế để chạy độc lập với các thiết bị khác nhau. Chính nhờ việc chuẩn hóa OID mà ta có thể dùng một SNMP application để lấy thông tin các loại device của các hãng khác nhau.

**Object Access**

Mỗi object có quyền truy cập là READ\_ONLY hoặc READ\_WRITE. Mọi object đều có thể đọc được nhưng chỉ những object có quyền READ\_WRITE mới có thể thay đổi được giá trị. VD : Tên của một thiết bị (sysName) là READ\_WRITE, ta có thể thay đổi tên của thiết bị thông qua giao thức SNMP. Tổng số port của thiết bị (ifNumber) là READ\_ONLY, dĩ nhiên ta không thể thay đổi số port của nó.

**Management Information Base**

MIB (cơ sở thông tin quản lý) là một cấu trúc dữ liệu gồm các đối tượng được quản lý (managed object), được dùng cho việc quản lý các thiết bị chạy trên nền TCP/IP. MIB là kiến trúc chung mà các giao thức quản

lý trên TCP/IP nên tuân theo, trong đó có SNMP. MIB được thể hiện thành 1 file (MIB file), và có thể biểu diễn thành 1 cây (MIB tree). MIB có thể được chuẩn hóa hoặc tự tạo.

Hình sau minh họa MIB tree :



Một node trong cây là một object, có thể được gọi bằng tên hoặc id. VD:

+ Node iso.org.dod.internet.mgmt.mib-2.system có OID là 1.3.6.1.2.1.1, chứa tất cả các object liên quan đến thông tin của một hệ thống như tên của thiết bị (iso.org.dod.internet.mgmt.mib-2.system.sysName hay 1.3.6.1.2.1.1.5).

+ Các OID của các hãng tự thiết kế nằm dưới tên định danh như sau: iso.org.dod.internet.private.enterprise. Ví dụ : Cisco nằm dưới iso.org.dod.internet.private.enterprise.cisco hay 1.3.6.1.4.1.9, Microsoft nằm dưới iso.org.dod.internet.private.enterprise.microsoft hay 1.3.6.1.4.1.311. Số 9 (Cisco) hay 311 (Microsoft) là số dành riêng cho các công ty do IANA cấp **5**. Nếu Cisco hay Microsoft chế tạo ra một thiết bị nào đó, thì thiết bị này có thể hỗ trợ các MIB chuẩn đã được định nghĩa sẵn (như mib-2) hay hỗ trợ MIB được thiết kế riêng. Các MIB được công ty nào thiết kế riêng thì phải nằm bên dưới OID của công ty đó.

Các objectID trong MIB được sắp xếp thứ tự nhưng không phải là liên tục, khi biết một OID thì không chắc chắn có thể xác định được OID tiếp theo trong MIB. VD trong chuẩn mib-2 6thì object ifSpecific và

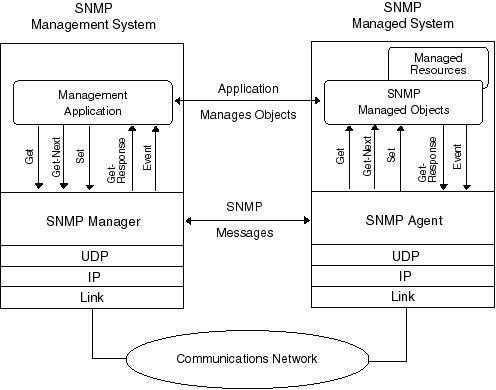
object atIfIndex nằm kề nhau nhưng OID lần lượt là 1.3.6.1.2.1.2.2.1.22 và 1.3.6.1.2.1.3.1.1.1.

Muốn hiểu được một OID nào đó thì bạn cần có file MIB mô tả OID đó. Một MIB file không nhất thiết phải chứa toàn bộ cây ở trên mà có thể chỉ chứa mô tả cho một nhánh con. Bất cứ nhánh con nào và tất cả lá của nó đều có thể gọi là một mib.

Một manager có thể quản lý được một device chỉ khi ứng dụng SNMP manager và ứng dụng SNMP agent cùng hỗ trợ một MIB. Các ứng dụng này cũng có thể hỗ trợ cùng lúc nhiều MIB.

1. Phương thức hoạt động

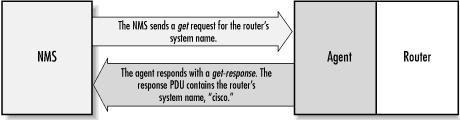
Hoạt động của SNMP theo mô hình sau:



Một số phương thức của SNMP: get, get-next, get-bulk (cho SNMP v2 và SNMP v3), set, get-response, trap (cảnh báo), notification (cho SNMP v2 và SNMP v3), inform (cho SNMP v2 và SNMP v3), report (cho SNMP v2 và SNMP v3).

* 1. Get

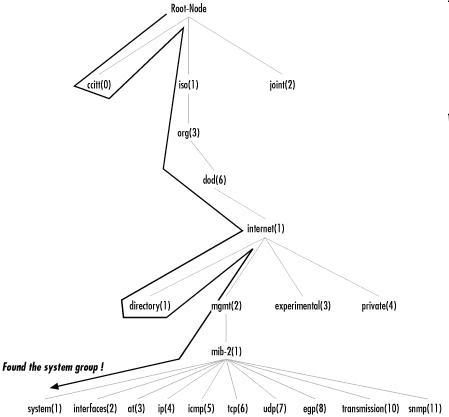
”Get” được gửi từ NMS yêu cầu tới agent. Agent nhận yêu cầu và xử lý với khả năng tốt nhất có thể. Nếu một thiết bị nào đó đang bận tải nặng, như router, nó không có khả năng trả lời yêu cầu nên nó sẽ hủy lời yêu cầu này. Nếu agent tập hợp đủ thông tin cần thiết cho lời yêu cầu, nó gửi lại cho NMS một ”get-response”:



Để agent hiểu được NMS cần tìm thông tin gì, nó dựa vào một mục trong ”get” là ”variable binding” hay varbind. Varbind là một danh sách các đối tượng của MIB mà NMS muốn lấy từ agent. Agent hiểu câu hỏi theo dạng: OID=value để tìm thông tin trả lời. Câu hỏi truy vấn cho trường hợp trong hình vẽ trên:  
*$ snmpget cisco.ora.com public .1.3.6.1.2.1.1.6.0  
system.sysLocation.0 = ""*  
Đây là một câu lệnh ”snmpget” trên Unix. ”cisco.ora.com” là tên của thiết bị, ”public” là chuổi chỉ đây là yêu cầu chỉ đọc (read-only), ”.1.3.6.1.2.1.1.6.0” là OID. ”.1.3.6.1.2.1.1” chỉ tới nhóm ”system” trong MIB. ”.6” chỉ tới một trường trong ”system” là ”sysLocation”. Trong câu lệnh này ta muốn hỏi Cisco router rằng việc định vị hệ thống đã được cài đặt chưa. Câu trả lời system.sysLocation.0 = "" tức là chưa cài đặt. Câu trả lời của ”snmpget” theo dạng của varbind: OID=value. Còn phần cuối trong OID ở ”snmpget”; ”.0” nằm trong quy ước của MIB. Khi hỏi một đối tượng trong MIB ta cần chỉ rõ 2 trường ”x.y’, ở đây là ”.6.0”. ”x” là OID thực tế của đối tượng. Còn ”.y” được dùng trong các đối tượng có hướng như một bảng để hiểu hàng nào của bảng, với trường hợp đối tượng vô hướng như trường hợp này ”y” = ”0”. Các hàng trong bảng được đánh số từ số 1 trở đi.  
Câu lệnh ”get” hữu ích trong việc truy vấn một đối tượng riêng lẻ trong MIB. Khi muốn biết thông tin về nhiều đối tượng thì ”get” tốn khá nhiều thời gian. Câu lệnh ’get-next” giải quyết được vấn đề này.

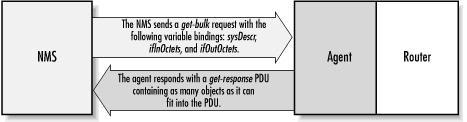
* 1. Get-next

”Get-next” đưa ra một dãy các lệnh để lấy thông tin từ một nhóm trong MIB. Agent sẽ lần lượt trả lời tất cả các đối tượng có trong câu truy vấn của ”get-next” tương tự như ”get”, cho đến khi nào hết các đối tượng trong dãy. Ví dụ ta dùng lệnh ”snmpwalk”. ”snmpwalk’ tương tự như ”snmpget’ nhưng không chỉ tới một đối tượng mà chỉ tới một nhánh nào đó:  
*$snmpwalk cisco.ora.com public system  
system.sysDescr.0 = "Cisco Internetwork Operating System Software   
..IOS (tm) 2500 Software (C2500-I-L), Version 11.2(5), RELEASE   
SOFTWARE (fc1)..Copyright (c) 1986-1997 by cisco Systems, Inc...  
Compiled Mon 31-Mar-97 19:53 by ckralik"  
system.sysObjectID.0 = OID: enterprises.9.1.19  
system.sysUpTime.0 = Timeticks: (27210723) 3 days, 3:35:07.23  
system.sysContact.0 = ""  
system.sysName.0 = "cisco.ora.com"  
system.sysLocation.0 = ""  
system.sysServices.0 = 6*  
Ở đây ta muôn lấy thông tin của nhóm ”system”, agent sẽ gửi trả toàn bộ thông tin của ”system” theo yêu cầu. Quá trình tìm nhóm ”system” trong MIB thực hiện theo cây từ gốc, đến một nút nếu có nhiều nhánh thì chọn nhánh tìm theo chỉ số của nhánh từ nhỏ đến lớn:



* 1. Get-bulk

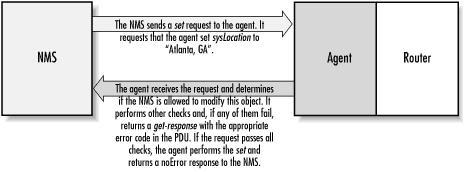
“Get-bulk” được định nghĩa trong SNMPv2. Nó cho phép lấy thông tin quản lý từ nhiều phần trong bảng. Dùng ”get” có thể làm được điều này. Tuy nhiên, kích thước của câu hỏi có thể bị giới hạn bởi agent. Khi đó nếu nó không thể trả lời toàn bộ yêu cầu, nó gửi trả một thông điệp lỗi mà không có dữ liệu. Với trường hợp dùng câu lệnh ”get-bulk”, agent sẽ gửi cang nhiều trả lời nếu nó có thể. Do đó, việc trả lời một phần của yêu cầu là có thể xảy ra. Hai trường cần khai báo trong ”get-bulk” là: ”nonrepeaters” và ”max-repetitions”. ”nonrepeaters” báo cho agent biết N đối tượng đầu tiên có thể trả lời lại như một câu lệnh ”get” đơn. ”mã-repeaters” báo cho agent biết cần cố gắng tăng lên tối đa M yêu cầu ”get-next” cho các đối tượng còn lại:



*$ snmpbulkget -v2c -B 1 3 linux.ora.com public sysDescr ifInOctets ifOutOctets  
system.sysDescr.0 = "Linux linux 2.2.5-15 #3 Thu May 27 19:33:18 EDT 1999 i686"  
interfaces.ifTable.ifEntry.ifInOctets.1 = 70840  
interfaces.ifTable.ifEntry.ifOutOctets.1 = 70840  
interfaces.ifTable.ifEntry.ifInOctets.2 = 143548020  
interfaces.ifTable.ifEntry.ifOutOctets.2 = 111725152  
interfaces.ifTable.ifEntry.ifInOctets.3 = 0  
interfaces.ifTable.ifEntry.ifOutOctets.3 = 0*  
  
ở đây, ta hỏi về 3 varbind: sysDescr, ifInOctets, và ifOutOctets. Tổng số varbind được tính theo công thức   
N + (M \* R)  
N: nonrepeater, tức số các đối tượng vô hướng  
M: max-repeatition  
R: số các đối tượng có hướng trong yêu cầu  
chỉ có sysDescr là vô hướng è N = 1  
M có thể đặt cho là 3 , tức là 3 trường cho mỗi ifInOctets và ifOutOctets. Có 2 đối tượng có hướng là ifInOctets và ifOutOctets è R = 2  
Tổng số có 1 + 3\*2 = 7 varbind  
Còn trường ”–v2c” là do ”get-bulk” là câu lệnh của SNMPv2 nên sử dụng ”-v2c” để chỉ rằng sử dụng PDU của SNMPv2. ”-B 1 3” là để đặt tham số N và M cho lệnh.

* 1. Set

Để thay đổi giá trị của một đối tượng hoặc thêm một hàng mới vào bảng. Đối tượng này cần phải được định nghĩa trong MIB là ”read-write” hay ”write-only”. NMS có thể dùng ”set’ để đặt giá trị cho nhiều đối tượng cùng một lúc:



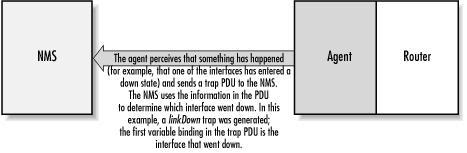
*$ snmpget cisco.ora.com public system.sysLocation.0  
system.sysLocation.0 = ""  
$ snmpset cisco.ora.com private system.sysLocation.0 s "Atlanta, GA"  
system.sysLocation.0 = "Atlanta, GA"  
$ snmpget cisco.ora.com public system.sysLocation.0  
system.sysLocation.0 = "Atlanta, GA"*  
  
Câu lệnh đầu là dung ”get” để lấy giá trị hiện tại của ”system.sysLocation”. Trong câu lệnh ”snmpset” các trường ”cisco.ora.com” và ”system.sysLocation.0” có ý nghĩa giống với ”get”. ”private” để chỉ đối tượng ”read-write’, và đặt giá trị mới bằng: ”s "Atlanta, GA"”. ”s” tức là đặt giá trị của ”system.sysLocation.0” thành string, và giá trị mới là "Atlanta, GA" . Varbind này được định nghĩa trong RFC 1213 là kiểu string tối đa 255 ký tự:  
*sysLocation OBJECT-TYPE  
SYNTAX DisplayString (SIZE (0..255))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"The physical location of this node (e.g., 'telephone closet,  
3rd floor')."  
::= { system 6 }*  
Có thể cài đặt nhiều đối tượng cùng lúc, tuy nhiên nếu có một hành động bị lỗi, toàn bộ sẽ bị hủy bỏ.

* 1. Error Response của ”get”, ”get-next”, ”get-bulk” và ”set”

Có nhiều loại lỗi báo lại từ agent:

* noError(0) Không có lỗi
* tooBig(1) Yêu cầu quá lớn để có thể dồn vào một câu trả lời.
* noSuchName(2) OID yêu cầu không tìm thấy, tức không tồn   
  tại ở agent.
* badValue(3) Câu lệnh “set” dùng không đúng với các   
  object “read-write” hay “write-only”
* readOnly(4) Lỗi này ít dùng. Lỗi “noSuchName” tương   
  đương với lỗi này.
* genErr(5) Dùng cho tất cả các lỗi còn lại, không nằm   
  trong các lỗi trên.
* Các loại lỗi của SNMPv1 mang tính chất chung nhất, không rõ ràng. Do đó SNMPv2 đưa ra thêm một số loại lỗi như sau:  
  + noAccess(6) Lỗi khi lệnh “set” cố gắng xâm nhập vào một biến cấm xâm nhập. Khi đó, biến đó có trường “ACCESS” là “not-accessible”  
  + wrongType(7) Lỗi xảy ra khi lệnh “set” đặt một kiểu dữ liệu khác với kiểu định nghĩa sẵn của đối tượng. Ví dụ khi “set” đặt giá trị kiểu string cho một đối tượng kiểu số nguyên INTEGER  
  wrongLength(8) Lỗi khi lệnh “set” đưa vào một giá trị có chiều dài lớn hơn chiều dài tối đa của đối tượng wrongEncoding(9) Lỗi khi lệnh “set” sử dụng cách mã hóa khác với cách đối tượng đã định nghĩa.  
  + wrongValue(10) Một biến được đặt một giá trị mà nó không   
  hiểu. Khi một biến theo kiểu liệt kê “enumeration” được đặt một giá trị không theo kiểu liệt kê.  
  + noCreation(11) Lỗi khi cố đặt một giá trị cho một biến   
  không tồn tại hoặc tạo một biến không có trong MIB   
  inconsistentValue Một biến MIB ở trạng thái không nhất   
  quán, và nó không chấp nhận bất cứ câu lệnh “set” nào.  
  resourceUnavailable(13) Không có tài nguyên hệ thống để thực hiện lệnh “set”  
  + commitFailed(14) Đại diện cho tất cả các lỗi khi lệnh   
  “set” thất bại  
  + undoFailed(15) Một lệnh “set” không thành công và agent   
  không thể phục hồi lại trạng thái trước khi lệnh “set” bắt đầu thất bại.  
  + authorizationError(16) Một lệnh SNMP không được xác thực, khi một người nào đó đưa ra mật mã không đúng.   
  + notWritable(17) Một biến không chấp nhận lệnh “set”  
  + inconsistentName(18) Cố gắng đặt một giá trị, nhưng việc cố   
  gắng thất bại vì biến đó đang ở tình trạng không nhất quán.
  1. SNMP Traps

Trap là cảnh báo của agent tự động gửi cho NMS để NMS biết có tình trạng xấu ở agent:



Khi nhận được một ”trap” từ agent, NMS không trả lời lại bằng ”ACK”. Do đó agent không thể nào biết được là lời cảnh báo của nó có tới được NMS hay không. Khi nhận được một ”trap” từ agent, nó tìm xem ”trap number” để hiểu ý nghĩa của ”trap” đó:

* coldStart (0) Thông báo agent vừa khởi động lại. Tất cả các   
  biến quản lý sẽ được reset, các biến kiểu “Counters” và “Gauges” được đặt về 0. “coldStart” dùng để xác định một thiết bị mới gia nhập vào mạng. Khi một thiết bị khởi động xong, nó gửi một “trap” tới NMS. Nếu địa chỉ NMS là đúng, NMS có   
  thể nhận được và xác định xem có quản lý thiết bị đó hay không.
* warmStart (1) Thông báo agent vừa khởi tạo lại, không có biến   
  nào bị reset.
* linkDown (2) Gửi đi khi một interface trên thiết bị chuyển   
  sang trạng thái “down”.
* linkUp (3) Gửi đi khi một interface trở lại trạng thái “up”.
* authenticationFailure (4): Cảnh báo khi một người nào đó cố truy cập vào agent đó mà không được xác thực.
* egpNeighborLoss (5): Cảnh báo một EGP lân cận bị “down”
* enterpriseSpecific (6): Đây là một “trap” riêng, chỉ được biết bởi agent và NMS tự định nghĩa riêng chúng. NMS sử dụng   
  phương pháp giải mã đặc biệt để hiểu được thông điệp này.  
  ”trap” được đưa ra trong MIB qua ”rdbmsOutOfSpace”:  
    
  *rdbmsOutOfSpace TRAP-TYPE  
  ENTERPRISE rdbmsTraps  
  VARIABLES { rdbmsSrvInfoDiskOutOfSpaces }  
  DESCRIPTION  
  "An rdbmsOutOfSpace trap signifies that one of the database servers managed by this agent has been unable to allocate space for one of the databases managed by this agent. Care should be taken to avoid flooding the network with these traps."  
  ::= 2*
  1. SNMP Notification

Để chuẩn hóa định dạng PDU ”trap” của SNMPv1 do PDU của ”get” và ”set” khác nhau, SNMPv2 đưa ra ”NOTIFICATION-TYPE”. Định dạng PDU của ”NOTIFICATION-TYPE” là để nhận ra ”get” và ”set”.   
  
*”NOTIFICATION-TYPE” được định nghĩa trong RFC 2863:  
linkDown NOTIFICATION-TYPE  
OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }  
STATUS current  
DESCRIPTION  
"A linkDown trap signifies that the SNMPv2 entity, acting in an agent role, has detected that the ifOperStatus object for one of its communication links left the down state and transitioned into some other state (but not into the notPresent state). This other state is indicated by the included value of ifOperStatus."  
::= { snmpTraps 3 }*

OID của “trap” này là 1.3.6.1.6.3.1.1.5.3, tức iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.sn mpMIBObjects.snmpTraps.linkDown.

* 1. SNMP Inform

SNMPv2 cung cấp cơ chế truyền thông giữa những NMS với nhau, gọi là SNMP inform. Khi một NMS gửi một SNMP inform cho một NMS khác, NMS nhận được sẽ gửi trả một ACK xác nhận sự kiện. Việc này giống với cơ chế của “get” và “set”.

* 1. SNMP Report

Được định nghĩa trong bản nháp của SNMPv2 nhưng không được phát triển. Sau đó được đưa vào SNMPv3 và hy vọng dùng để truyền thông giữa các hệ thống SNMP với nhau.

1. Triển khai mô hình thực tế