HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

PROJECT 1

COLLECTION SNMP DATA

Đặng Minh Đức
duc.dm214956@sis.hust.edu.vn

Đỗ Mạnh Cường
cuong.dm214948@sis.hust.edu.vn

Major: Cyber Security

Supervisor: Associate Professor Nguyễn Quốc Khánh _____

Signature

Department: Cyber Security

School: School of Information and Communications Technology

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

1.1 Problem Statement

Lập trình chương trình thu thập dữ liệu snmp từ thiết bị mạng và máy chủ.

1.2 Target

Mục tiêu của đồ án chương trình thu thập dữ liệu SNMP là thu thập thông tin từ các thiết bị mạng sử dụng giao thức SNMP (Simple Network Management Protocol) để giúp quản trị viên mạng có thể theo dõi và quản lý tình trạng hoạt động của hệ thống mạng một cách hiệu quả.

1.3 Contributions

Chương trình được thiết kế để thu thập các thông tin về tài nguyên hệ thống như CPU, bộ nhớ, băng thông, tình trạng hoạt động của các cổng mạng và các thông tin khác liên quan đến hệ thống mạng. Sau đó, thông tin này sẽ được xử lý và lưu trữ trong cơ sở dữ liệu để quản trị viên có thể dễ dàng truy cập và theo dõi.

1.4 Organization of Thesis

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về lý thuyết cần biết về snmp để xây dựng chương trình

Chương 3 trình bày về ứng dụng và cách hoạt động.

Chương 4 trình bày về kết luận và hướng phát triển.

chương 5 nêu những tài liệu đã tham khảo.

CHAPTER 2. THEORY

SNMP là "giao thức quản lý mạng đơn giản", dịch từ cụm từ "Simple Network Management Protocol". Thế nào là giao thức quản lý mạng đơn giản ?

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.

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.

Ưu điểm trong thiết kế của SNMP:

SNMP được thiết kế để đơn giản hóa quá trình quản lý các thành phần trong mạng. Nhờ đó các phần mềm SNMP có thể được phát triển nhanh và tốn ít chi phí (trong chương 5 tác giả sẽ trình bày cách xây dựng phần mềm giám sát SNMP, bạn sẽ thấy tính đơn giản của nó).

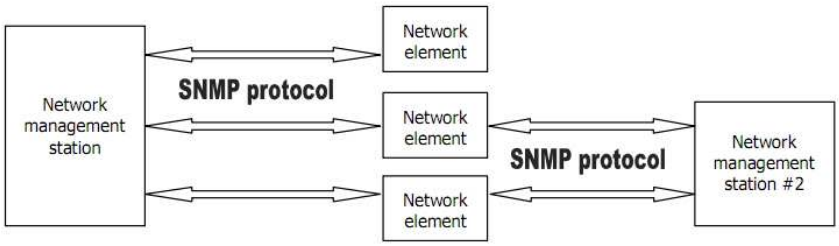
SNMP được thiết kế để có thể mở rộng các chức năng quản lý, giám sát. Không

có giới hạn rằng SNMP có thể quản lý được cái gì. Khi có một thiết bị mới với các thuộc tính, tính năng mới thì người ta có thể thiết kế "custom" SNMP để phục vụ cho riêng mình (trong chương 3 tác giả sẽ trình bày file cấu trúc dữ liệu của SNMP).

SNMP được thiết kế để có thể hoạt động độc lập với các kiến trúc và cơ chế của các thiết bị hỗ trợ SNMP. Các thiết bị khác nhau có hoạt động khác nhau nhưng đáp ứng SNMP là giống nhau. VD bạn có thể dùng 1 phần mềm để theo dõi dung lượng ổ cứng còn trống của các máy chủ chạy HĐH Windows và Linux; trong khi nếu không dùng SNMP mà làm trực tiếp trên các HĐH này thì bạn phải thực hiện theo các cách khác nhau.

Các thành phần trong SNMP:

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.

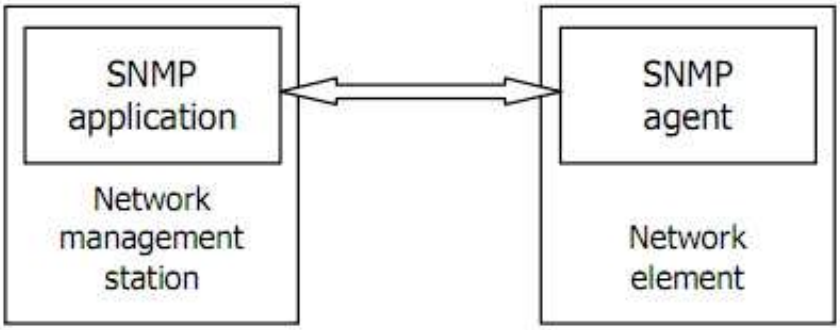


Network element là các thiết bị, máy tính, hoặc phần mềm tương thích SNMP và được quản lý bởi network management station. Như vậy element bao gồm device, host và application. Một management station có thể quản lý nhiều element, một element cũng có thể được quản lý bởi nhiều management station. Vậy nếu 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. Ví dụ :

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.

Bạn hãy khoan thắc mắc ý nghĩa của từng chữ số trong OID, chúng sẽ được giải thích trong phần sau. Một object chỉ có một OID, chẳng hạn tên của thiết bị là một object. Tuy nhiên nếu một thiết bị lại có nhiều tên thì làm thế nào để phân biệt ? Lúc này người ta dùng thêm 1 chỉ số gọi là “scalar instance index” (cũng có thể gọi là “sub-id”) đặt ngay sau OID. Ví dụ :

+Tên thiết bị được gọi là sysName, OID là 1.3.6.1.2.1.1.5; nếu thiết bị có 2 tên thì chúng sẽ được gọi là sysName.0 and sysName.1 và có OID lần lượt là 1.3.6.1.2.1.1.5.0 and 1.3.6.1.2.1.1.5.1.

+Địa chỉ Mac address được gọi là ifPhysAddress, OID là 1.3.6.1.2.1.2.2.1.6; nếu

thiết bị có 2 mac address thì chúng sẽ được gọi là ifPhysAddress.0 and ifPhysAddress.1 và có OID lần lượt là 1.3.6.1.2.1.2.2.1.6.0 and 1.3.6.1.2.1.2.2.1.6.1.

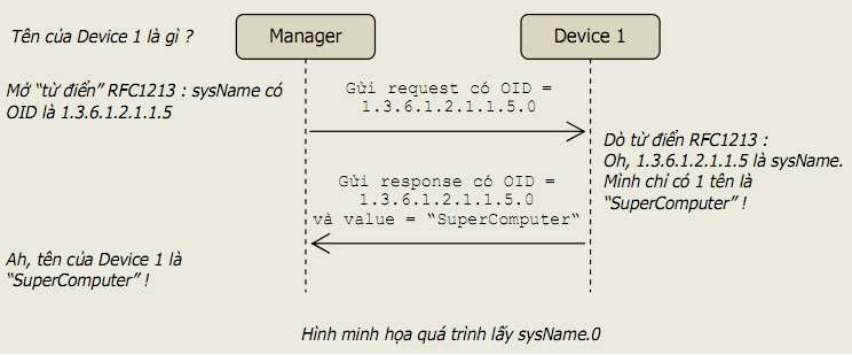
+ Tổng số port được gọi là ifNumber, giá trị này chỉ có 1 (duy nhất) nên OID của nó không có phân cấp con và vẫn là 1.3.6.1.2.1.2.1.

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. Bạn 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 chúng 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 bạn 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.



Hình minh họa quá trình lấy sysName.0

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

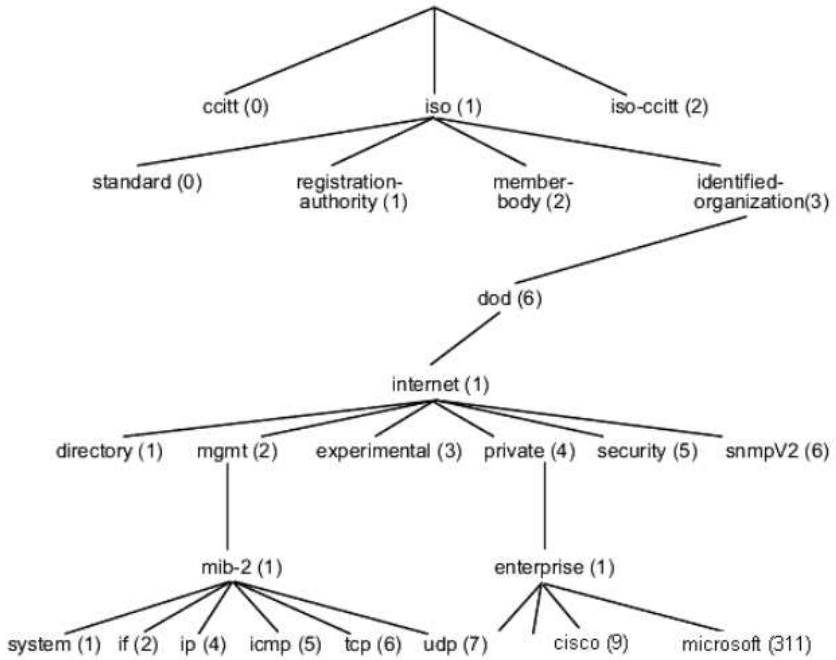
Mỗi object có quyền truy cập là READ ONLY hoặc READ WRITE. Mọi object đều có thể đọc được nhưng

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. Ví dụ :

+ 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 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. 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 thì 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.

Các phương thức của SNMP

Giao thức SNMPv1 có 5 phương thức hoạt động, tương ứng với 5 loại bản tin như sau :

Bản tin/phương thức	Mô tả tác dụng
GetRequest	Manager gửi GetRequest cho agent để yêu cầu agent cung cấp thông tin nào đó dựa vào ObjectID (trong GetRequest có chứa OID)
GetNextRequest	Manager gửi GetNextRequest có chứa một ObjectID cho agent để yêu cầu cung cấp thông tin nằm kế tiếp ObjectID đó trong MIB.
SetRequest	Manager gửi SetRequest cho agent để đặt giá trị cho đối tượng của agent dựa vào ObjectID.
GetResponse	Agent gửi GetResponse cho Manager để trả lời khi nhận được GetRequest/GetNextRequest
Trap	Agent tự động gửi Trap cho Manager khi có một sự kiện xảy ra đối với một object nào đó trong agent.

Mỗi bản tin đều có chứa OID để cho biết object mang trong nó là gì. OID trong GetRequest cho biết nó muốn lấy thông tin của object nào. OID trong GetResponse cho biết nó mang giá trị của object nào. OID trong SetRequest chỉ ra nó muốn thiết lập giá trị cho object nào. OID trong Trap chỉ ra nó thông báo sự kiện xảy ra đối với object nào.

GetRequest

Bản tin GetRequest được manager gửi đến agent để lấy một thông tin nào đó. Trong GetRequest có chứa OID của object muốn lấy.

Trong một bản tin GetRequest có thể chứa nhiều OID, nghĩa là dùng một GetRequest có thể lấy về cùng lúc nhiều thông tin.

GetNextRequest

Bản tin GetNextRequest cũng dùng để lấy thông tin và cũng có chứa OID, tuy nhiên nó dùng để lấy thông tin của object nằm kế tiếp object được chỉ ra trong bản

tin.

Tại sao phải có phương thức GetNextRequest ? Như bạn đã biết khi đọc qua những phần trên : một MIB bao gồm nhiều OID được sắp xếp thứ tự nhưng không liên tục, nếu biết một OID thì không xác định được OID kế tiếp. Do đó ta cần GetNextRequest để lấy về giá trị của OID kế tiếp. Nếu thực hiện GetNextRequest liên tục thì ta sẽ lấy được toàn bộ thông tin của agent.

SetRequest

Bản tin SetRequest được manager gửi cho agent để thiết lập giá trị cho một object nào đó. Ví dụ :

+ Có thể đặt lại tên của một máy tính hay router bằng phần mềm SNMP manager, bằng cách gửi bản tin SetRequest có OID là 1.3.6.1.2.1.1.5.0 (sysName.0) và có giá trị là tên mới cần đặt.

+ Có thể shutdown một port trên switch bằng phần mềm SNMP manager, bằng cách gửi bản tin có

OID là 1.3.6.1.2.1.2.2.1.7 (ifAdminStatus) và có giá trị là 2

* ifAdminStatus có thể mang 3 giá trị là UP (1), DOWN (2) và TESTING (3).

Chỉ những object có quyền READ WRITE mới có thể thay đổi được giá trị.

GetResponse

Mỗi khi SNMP agent nhận được các bản tin GetRequest, GetNextRequest hay SetRequest thì nó sẽ gửi lại bản tin GetResponse để trả lời. Trong bản tin GetResponse có chứa OID của object được request và giá trị của object đó.

Trap

Bản tin Trap được agent tự động gửi cho manager mỗi khi có sự kiện xảy ra bên trong agent, các sự kiện này không phải là các hoạt động thường xuyên của agent mà là các sự kiện mang tính biến cố. Ví dụ : Khi có một port down, khi có một người dùng login không thành công, hoặc khi thiết bị khởi động lại, agent sẽ gửi trap cho manager.

Tuy nhiên không phải mọi biến cố đều được agent gửi trap, cũng không phải mọi agent đều gửi trap khi xảy ra cùng một biến cố. Việc agent gửi hay không gửi trap cho biến cố nào là do hãng sản xuất device/agent quy định.

Phương thức trap là độc lập với các phương thức request/response. SNMP request/response dùng để quản lý còn SNMP trap dùng để cảnh báo. Nguồn gửi trap gọi là Trap Sender và nơi nhận trap gọi là Trap Receiver. Một trap sender có thể được cấu hình

để gửi trap đến nhiều trap receiver cùng lúc.

Có 2 loại trap : trap phổ biến (generic trap) và trap đặc thù (specific trap). Generic trap được quy định trong các chuẩn SNMP, còn specific trap do người dùng tự định nghĩa (người dùng ở đây là hãng sản xuất SNMP device). Loại trap là một số nguyên chứa trong bản tin trap, dựa vào đó mà phía nhận trap biết bản tin trap có nghĩa gì.

Theo SNMPv1, generic trap có 7 loại sau : coldStart(0), warmStart(1), linkDown(2), linkUp(3), authenticationFailure(4), egpNeighborloss(5), enterpriseSpecific(6). Giá trị trong ngoặc là mã số của các loại trap. Ý nghĩa của các bản tin generic-trap như sau :

+ coldStart : thông báo rằng thiết bị gửi bản tin này đang khởi động lại (reinitialize) và cấu hình của nó có thể bị thay đổi sau khi khởi động.

+ warmStart : thông báo rằng thiết bị gửi bản tin này đang khởi động lại và giữ nguyên cấu hình cũ.

+ linkDown : thông báo rằng thiết bị gửi bản tin này phát hiện được một trong những kết nối truyền thông (communication link) của nó gặp lỗi. Trong bản tin trap có tham số chỉ ra ifIndex của kết nối bị lỗi.

+ linkUp : thông báo rằng thiết bị gửi bản tin này phát hiện được một trong những kết nối truyền thông của nó đã khôi phục trở lại. Trong bản tin trap có tham số chỉ ra ifIndex của kết nối được khôi phục.

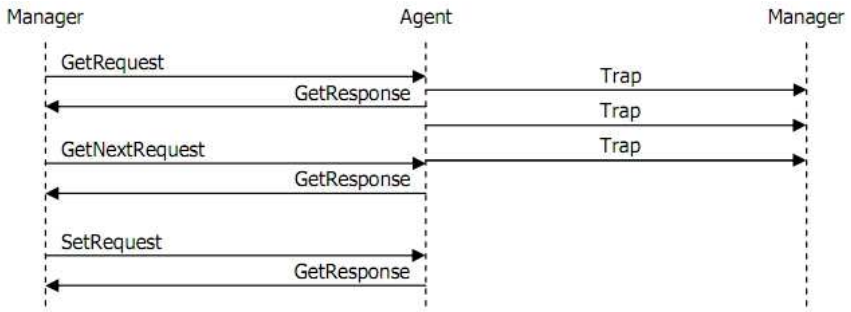
+ authenticationFailure : thông báo rằng thiết bị gửi bản tin này đã nhận được một bản tin không được chứng thực thành công (bản tin bị chứng thực không thành công có thể thuộc nhiều giao thức khác nhau như telnet, ssh, snmp, ftp, . . .). Thông thường trap loại này xảy ra là do user đăng nhập không thành công vào thiết bị.

+ egpNeighborloss : thông báo rằng một trong số những “EGP neighbor” của thiết bị gửi trap đã bị coi là down và quan hệ đối tác (peer relationship) giữa 2 bên không còn được duy trì.

+ enterpriseSpecific : thông báo rằng bản tin trap này không thuộc các kiểu generic như trên mà nó là một loại bản tin do người dùng tự định nghĩa.

Người dùng có thể tự định nghĩa thêm các loại trap để làm phong phú thêm khả năng cảnh báo của thiết bị như : boardFailed, configChanged, powerLoss, cpuTooHigh, v.v. . . . Người dùng tự quy định ý nghĩa và giá trị của các specific trap này, và dĩ nhiên chỉ những trap receiver và trap sender hỗ trợ cùng một MIB mới có thể hiểu ý nghĩa của specific trap. Do đó nếu bạn dùng một phần mềm trap

receiver bất kỳ để nhận trap của các trap sender bất kỳ, bạn có thể đọc và hiểu các generic trap khi chúng xảy ra; nhưng bạn sẽ không hiểu ý nghĩa các specific trap khi chúng hiện lên màn hình vì bản tin trap chỉ chứa những con số.



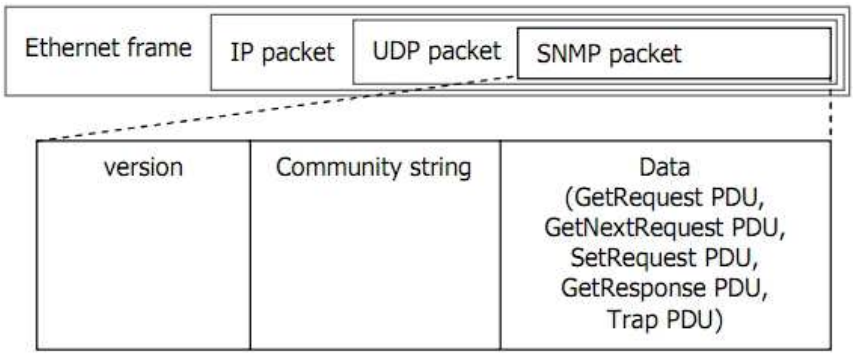
Hình minh họa các phương thức của SNMPv1

Đối với các phương thức Get/Set/Response thì SNMP Agent lắng nghe ở port UDP 161, còn phương thức trap thì SNMP Trap Receiver lắng nghe ở port UDP 162.

SNMPwalk là một công cụ dòng lệnh được sử dụng để truy vấn thông tin từ các thiết bị hỗ trợ SNMP. Nó hoạt động bằng cách gửi các yêu cầu SNMP GETNEXT tới các đối tượng (object) trên thiết bị và hiển thị kết quả trả về. Các đối tượng này được tổ chức thành một cây đối tượng (object tree) và mỗi đối tượng có một định danh duy nhất trong cây. Ví dụ, bạn có thể sử dụng SNMPwalk để truy vấn thông tin về các giao diện mạng trên một router, bao gồm địa chỉ IP, tốc độ truyền dữ liệu, trạng thái hoạt động, và các thông số khác. Công cụ này rất hữu ích trong việc giám sát và quản lý các thiết bị mạng.

SNMPtable là một công cụ dòng lệnh trong giao thức SNMP (Simple Network Management Protocol) được sử dụng để hiển thị dữ liệu bảng của các thiết bị mạng. Cụ thể, snmptable cho phép bạn lấy dữ liệu từ một bảng SNMP và hiển thị nó dưới dạng một bảng ASCII. Với snmptable, bạn có thể lấy thông tin từ các bảng SNMP như bảng định tuyến, bảng ARP, bảng MAC, v.v. Các thông tin này đều được sắp xếp theo cách chính xác và có thể được lọc hoặc sắp xếp lại theo nhu cầu sử dụng. Kết quả trả về từ snmptable thường được hiển thị dưới dạng một bảng gồm các cột và hàng, với mỗi hàng tương ứng với một bản ghi trong bảng SNMP. Các đối tượng SNMP trong bảng được lưu trữ dưới dạng OID (Object Identifier), và các giá trị của chúng được hiển thị dưới dạng chuỗi ASCII hoặc số. Snmptable là một công cụ hữu ích cho việc kiểm tra và giám sát các thiết bị mạng, đặc biệt là trong môi trường mạng lớn với nhiều thiết bị và các bảng SNMP phức tạp. Cấu trúc bản tin SNMP

SNMP chạy trên nền UDP. Cấu trúc của một bản tin SNMP bao gồm : version, community và data.



+ Version : v1 = 0, v2c = 1, v2u = 2, v3 = 3.

+ Phần Data trong bản tin SNMP gọi là PDU (Protocol Data Unit). SNMPv1 có 5 phương thức hoạt động tương ứng 5 loại PDU. Tuy nhiên chỉ có 2 loại định dạng bản tin là PDU và Trap-PDU; trong đó các bản tin Get, GetNext, Set, GetResponse có cùng định dạng là PDU, còn bản tin Trap có định dạng là Trap-PDU.

CHAPTER 3. APPLICATION

3.1 Overview

Chương trình bao gồm GET, GETNEXT, WALK cho phép người dùng lấy dữ liệu từ máy chủ và thiết bị mạng và hiện ra OID,tên, mô tả,...

3.2 Phương thức GET, GETNEXT, WALK

- Khi thực hiện yêu cầu GET, chúng ta cần chỉ định địa chỉ của thiết bị mạng cần truy vấn và OID (Object Identifier) của biến mạng mà chúng ta muốn lấy giá trị. Mỗi biến trong hệ thống mạng được định danh bằng một OID duy nhất. Sau khi nhận được yêu cầu GET, thiết bị mạng sẽ trả về giá trị hiện tại của biến mạng được yêu cầu. Nếu không có lỗi xảy ra, giá trị sẽ được trả về trong phản hồi của thiết bị mạng. Yêu cầu GET là một trong những yêu cầu cơ bản của giao thức SNMP và được sử dụng rộng rãi trong các ứng dụng quản lý mạng để lấy thông tin từ các thiết bị mạng.

- Yêu cầu GETNEXT được sử dụng để lấy giá trị của biến mạng tiếp theo trong hệ thống mạng.Khi thực hiện yêu cầu GETNEXT, chúng ta cần chỉ định địa chỉ của thiết bị mạng cần truy vấn và OID (Object Identifier) của biến mạng hiện tại. Thiết bị mạng sẽ trả về giá trị của biến mạng tiếp theo trong danh sách biến mạng được lưu trữ trên thiết bị mạng. Yêu cầu GETNEXT được sử dụng trong các trường hợp khi người dùng muốn lấy giá trị của tất cả các biến mạng trong hệ thống mạng, hoặc khi người dùng không biết chính xác OID của biến mạng cần lấy. Yêu cầu GETNEXT trả về giá trị đầu tiên của biến mạng tiếp theo trong danh sách biến mạng được lưu trữ trên thiết bị mạng. Nếu không có biến mạng nào khác trong danh sách, thiết bị mạng sẽ trả về một mã lỗi. Yêu cầu GETNEXT là một trong những yêu cầu cơ bản của giao thức SNMP và được sử dụng rộng rãi trong các ứng dụng quản lý mạng để lấy thông tin từ các thiết bị mạng.

- WALK là một trong những yêu cầu được sử dụng để lấy thông tin về tất cả các biến mạng trong một nhóm biến liên quan đến nhau. Yêu cầu WALK được sử dụng để lấy giá trị của tất cả các biến mạng trong một nhóm biến mạng liên quan đến nhau, thay vì chỉ lấy giá trị của một biến mạng cụ thể. Khi thực hiện yêu cầu WALK, chúng ta cần chỉ định địa chỉ của thiết bị mạng cần truy vấn và OID (Object Identifier) của nhóm biến mạng mà chúng ta muốn lấy giá trị. Thiết bị mạng sẽ trả về giá trị của tất cả các biến mạng trong nhóm biến mạng đó. Yêu cầu WALK được sử dụng rộng rãi trong các ứng dụng quản lý mạng để lấy thông tin về tất cả các biến mạng trong một nhóm biến liên quan đến nhau. Việc sử dụng yêu cầu WALK thay vì lấy giá trị của từng biến mạng một cách độc lập giúp tiết kiệm thời gian và

nâng cao hiệu quả trong việc thu thập thông tin. Tuy nhiên, việc sử dụng yêu cầu WALK có thể tốn nhiều tài nguyên mạng và có thể gây tắc nghẽn mạng nếu lượng dữ liệu truy vấn là quá lớn. Do đó, việc sử dụng yêu cầu WALK cần được thực hiện cẩn thận và tính toán để đảm bảo không ảnh hưởng đến hiệu suất hoạt động của hệ thống mạng.

- Cả 3 phương thức đều được lập trình bằng ngôn ngữ java sử dụng thư viện snmp4j.org

3.3 Hiện thông tin

Khi người dùng nhập ip và chọn phương thức thì chương trình sẽ in ra thông tin những gì nhận được : OID, name, description, status, maxaccess,... lấy từ file json parse từ file mib sử dụng tool mibdump của python.

- MIB (Management Information Base) file là một tập tin văn bản được sử dụng để định nghĩa và mô tả các đối tượng quản lý (managed objects) trong hệ thống mạng. MIB file chứa các thông tin về tên, OID (Object Identifier), mô tả, đơn vị đo lường, phạm vi giá trị và các thuộc tính khác của các đối tượng quản lý. MIB file được sử dụng để truyền thông tin giữa các thiết bị mạng và các ứng dụng quản lý mạng. Các ứng dụng quản lý mạng sử dụng các thông tin trong MIB file để hiểu các đối tượng quản lý trên các thiết bị mạng, và để thực hiện các hoạt động quản lý như giám sát, cấu hình và quản lý lỗi.

Một số định dạng MIB file phổ biến bao gồm: SNMPv1 MIB file, SNMPv2 MIB file, SMIv1 MIB file và SMIv2 MIB file. Các định dạng này có các cú pháp và cấu trúc khác nhau, tuy nhiên, chúng đều định nghĩa các đối tượng quản lý và các thuộc tính của chúng dưới dạng danh sách hoặc cây phân cấp. Việc định nghĩa các MIB file giúp cho việc quản lý mạng trở nên dễ dàng và hiệu quả hơn. Nó giúp cho các ứng dụng quản lý mạng hiểu được các đối tượng quản lý trên các thiết bị mạng và đảm bảo tính nhất quán trong việc quản lý mạng.

- OID (Object Identifier) là một chuỗi các số duy nhất được sử dụng để xác định một đối tượng quản lý (managed object) trong MIB (Management Information Base) của một hệ thống mạng. Mỗi đối tượng quản lý trong MIB đều có một OID duy nhất để xác định nó. OID được sử dụng để định danh các đối tượng quản lý trong MIB. Nó là một chuỗi các số nguyên (integer) được phân tách bằng dấu chấm (.). Ví dụ, OID của đối tượng quản lý "ifInDiscards" trong IF-MIB là "1.3.6.1.2.1.2.2.1.13". Mỗi số trong OID đại diện cho một nhóm đối tượng quản lý trong cây phân cấp OID. Các nhóm này được định nghĩa trong các chuẩn MIB khác nhau, và mỗi chuẩn MIB đều có một OID cơ sở riêng để bắt đầu phân cấp. Ví dụ, OID cơ sở của IF-MIB là "1.3.6.1.2.1.2". Việc sử dụng OID giúp cho việc truy

cập và quản lý các đối tượng trong MIB trở nên dễ dàng và hiệu quả hơn. Nó cũng giúp đảm bảo tính duy nhất và nhất quán trong việc định danh các đối tượng quản lý trong MIB.

"Name" là một thuộc tính được định nghĩa trong các đối tượng quản lý (managed object) trong MIB (Management Information Base), để xác định tên của đối tượng quản lý đó. Thuộc tính "name" được sử dụng để đặt tên cho các đối tượng quản lý, giúp cho việc nhận biết và sử dụng các đối tượng quản lý trở nên dễ dàng hơn. Tên của đối tượng quản lý có thể được sử dụng để truy cập và thao tác với đối tượng quản lý trong các ứng dụng quản lý mạng. Việc định nghĩa thuộc tính "name" giúp cho việc quản lý mạng trở nên dễ dàng hơn và đảm bảo tính nhất quán trong việc đặt tên các đối tượng quản lý. Nó giúp cho người dùng và nhà phát triển hoàn toàn hiểu được tên của các đối tượng quản lý, từ đó sử dụng chúng một cách dễ dàng và hiệu quả.

- "Description" là một thuộc tính được định nghĩa trong các đối tượng quản lý (managed object) trong MIB (Management Information Base), để mô tả một cách chi tiết và rõ ràng về chức năng, ý nghĩa và cách sử dụng của một đối tượng quản lý. Thuộc tính "description" được sử dụng để cung cấp thông tin cần thiết về đối tượng quản lý, giúp người dùng hiểu rõ hơn về chức năng của đối tượng đó và cách sử dụng nó. Mô tả có thể bao gồm các thông tin về đơn vị đo lường, phạm vi giá trị, ràng buộc và các thông tin khác liên quan đến đối tượng quản lý. Việc định nghĩa thuộc tính "description" giúp cho việc quản lý mạng trở nên dễ dàng hơn và đảm bảo tính nhất quán trong việc sử dụng các đối tượng quản lý. Nó giúp cho người dùng và nhà phát triển hoàn toàn hiểu được chức năng và ý nghĩa của các đối tượng quản lý, từ đó sử dụng chúng một cách đúng đắn và hiệu quả.

- "maxaccess" là một thuộc tính được định nghĩa trong các đối tượng quản lý (managed object) trong MIB (Management Information Base), để xác định quyền truy cập tối đa (maximum access) mà một đối tượng quản lý có thể có. Các giá trị của thuộc tính "maxaccess" có thể bao gồm "not-accessible" (không thể truy cập), "read-only" (chỉ đọc), "read-write" (đọc và ghi), "write-only" (chỉ ghi), "read-create" (đọc và tạo mới), "accessible-for- notify" (truy cập cho thông báo). Giá trị của thuộc tính "maxaccess" xác định quyền truy cập mà các ứng dụng quản lý có thể có để truy cập và thay đổi giá trị của đối tượng quản lý. Việc định nghĩa thuộc tính "maxaccess" giúp cho việc quản lý mạng được an toàn và được kiểm soát chặt chẽ hơn. Nó giúp ngăn chặn việc truy cập và thay đổi không đúng của các đối tượng quản lý, đồng thời cũng giúp đơn giản hóa việc phát triển ứng dụng quản lý mạng bởi việc định rõ quyền truy cập tối đa của mỗi đối tượng quản lý.

- "Status" là một thuộc tính được định nghĩa trong các đối tượng quản lý trong MIB (Management Information Base) để chỉ trạng thái của đối tượng đó. Thuộc tính này mô tả tình trạng hiện tại của đối tượng quản lý, xác định liệu nó có được sử dụng hay không và có được hỗ trợ trong phiên bản hiện tại của MIB hay không. Các giá trị của thuộc tính "status" có thể bao gồm "current" (hiện tại), "obsolete" (lỗi thời), "deprecated" (không còn được khuyến cáo sử dụng) và "experimental" (thử nghiệm). Nếu một đối tượng quản lý có trạng thái là "current", nó được hỗ trợ trong phiên bản hiện tại của MIB và có thể được sử dụng. Nếu đối tượng quản lý có trạng thái là "obsolete", nó đã bị thay thế bởi một đối tượng khác và không nên được sử dụng nữa. Thuộc tính "status" có thể giúp người quản trị mạng biết được xem đối tượng quản lý đó có hỗ trợ trong phiên bản hiện tại của MIB hay không, và hướng dẫn họ sử dụng đối tượng quản lý đó một cách đúng đắn.

CHAPTER 4. CONCLUSIONS

4.1 Summary

Chương trình thu thập SNMP (Simple Network Management Protocol) được sử dụng để thu thập thông tin về các thiết bị mạng như router, switch, firewall, server và các thiết bị mạng khác. Chương trình này có thể thu thập các thông tin như:

1. Thông tin về tình trạng hoạt động của các thiết bị mạng, bao gồm thông tin về tình trạng kết nối, tình trạng bộ nhớ và tình trạng CPU.

2. Thông tin về các giao thức mạng đang hoạt động, bao gồm thông tin về các cổng mạng đang sử dụng, thống kê lưu lượng mạng và thông tin về các giao thức mạng khác.

3. Thông tin về các thiết bị mạng, bao gồm thông tin về cấu hình thiết bị, địa chỉ IP, tên miền, tên thiết bị và các thông tin khác.

5. Thông tin về các thiết bị mạng ngoại vi, bao gồm thông tin về các thiết bị như máy in, máy quét và các thiết bị khác được kết nối với mạng.

Chương trình thu thập SNMP là một công cụ quan trọng để giám sát và quản lý hệ thống mạng, giúp người quản trị mạng có thể dễ dàng theo dõi tình trạng hoạt động của các thiết bị mạng và phát hiện các sự cố hoặc vấn đề trên mạng.

Chương trình hiện tại vẫn còn chưa có khả năng Thông tin về các sự kiện và cảnh báo, bao gồm thông tin về các sự kiện khắc phục sự cố, cảnh báo an ninh và các sự kiện khác, thông tin về các ứng dụng và dịch vụ đang chạy trên các thiết bị mạng, bao gồm thông tin về các ứng dụng web, cơ sở dữ liệu và các dịch vụ khác, thông tin về các thiết bị mạng ngoại vi, bao gồm thông tin về các thiết bị như máy in, máy quét và các thiết bị khác được kết nối với mạng.

4.2 Suggestion for Future Works

Bọn em sẽ phát triển thêm trap để thêm tính năng báo cáo về một sự kiện hoặc trạng thái cụ thể, được sử dụng để cảnh báo người quản trị mạng về các sự kiện quan trọng trên mạng, như lỗi hệ thống, tình trạng mạng kém, thay đổi cấu hình, hoặc các hoạt động không bình thường khác.

Khi một sự kiện xảy ra trên một thiết bị mạng, thiết bị sẽ gửi một trap tới trạm quản lý SNMP. Trap này chứa thông tin về sự kiện, bao gồm cả thông tin về thiết bị gửi trap. Trong khi các yêu cầu SNMP (SNMP request) được gửi từ trạm quản lý tới các thiết bị mạng để yêu cầu thông tin, trap được gửi từ các thiết bị mạng tới trạm quản lý SNMP mà không có yêu cầu trước.

Trong quá trình quản lý mạng, trap là một công cụ hữu ích để cảnh báo người quản trị mạng về các sự kiện quan trọng trên mạng một cách nhanh chóng. Các người quản trị mạng có thể cấu hình trạm quản lý SNMP để xử lý các trap một cách tự động, ví dụ như gửi thông báo qua email hoặc tin nhắn khi nhận được trap về một sự kiện quan trọng nhất định.

CHAPTER 5. REFERENCE

https://ireasoning.com/mibbrowser.shtml

https://www.snmp4j.org/

https://github.com/eozer/awesome-snmp

https://www.circitor.fr/Mibs/Mibs.php

https://github.com/micmiu/snmp-tutorial/blob/master/snmp4j-1x-demo/src/main/java/com/micmiu