ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

GIÁM SÁT MẠNG

Giảng viên: Lê Tự Thanh
Email : ltthanh@vku.udn.vn
Website : www.vku.udn.vn

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

3.1. Khái niệm lập trình SNMP

Trong thực tế có nhiều thiết bị, ứng dụng được các hãng thiết kế MIB riêng, người giám sát mạng không thể giám sát chúng bằng ứng dụng SNMP thông thường. Người giám sát mạng có thể dùng phần mềm của chính hãng thiết bị đó để giám sát. Nhưng nếu có nhiều chủng loại thiết bị khác nhau thì bắt buộc người giám sát mạng phải dùng từng phần mềm riêng biệt khác nhau.

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

3.1. Khái niệm lập trình SNMP

Vấn đề đặt ra là làm thế nào để có thể dùng một ứng dụng duy nhất để giám sát tất cả chúng ? Lúc này bạn cần biết cách lập trình ứng dụng giám sát SNMP.

Cũng có một số phần mềm cho phép giám sát “custom mib” nhưng phần lớn chưa đáp ứng hoàn toàn nhu cầu của bạn. Các thiết bị gửi các event dạng trap đến một trap host. Định nghĩa trap chuẩn chỉ có một số event rất ít, do đó các dòng sản phẩm khác nhau đều có định nghĩa rất nhiều trap enterpriseSpecific mà phải dùng sản phẩm của chính hãng mới có thể đọc được.

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

3.1. Khái niệm lập trình SNMP

Nếu bạn có file mib mô tả event của các thiết bị, làm thế nào để dùng một ứng dụng duy nhất để làm host nhận event và cảnh báo cho tất cả các chủng loại thiết bị ? Lúc này bạn cần biết cách lập trình ứng dụng SNMP Trap receiver.

Giả sử bạn viết một ứng dụng nào đó, ứng dụng này chạy trên rất nhiều server. Người quản trị cần giám sát hiệu năng ứng dụng của bạn trên tất cả các server mà không cần phải truy cập vào từng server để lấy thông tin.

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

3.1. Khái niệm lập trình SNMP

Bạn có thể thiết kế giao thức và phần mềm giám sát riêng, nhưng nếu sử dụng SNMP thì người dùng có thể dùng các phần mềm có sẵn tính năng “custom mib” như Solarwinds để giám sát ứng dụng của bạn. Lúc này bạn cần biết cách lập trình ứng dụng SNMP Agent để bổ sung tính năng này vào ứng dụng của bạn.

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

3.1. Khái niệm lập trình SNMP

Trước khi sử dụng lệnh SNMP, hãy đảm bảo cài đặt các tệp ILOM MIB.

http://net-snmp.sourceforge.net/wiki/index.php/

https://docs.oracle.com/cd/E19201-01/820-6413-13/SNMP_commands_reference_appendix.html

Lệnh snmpget

snmpget -mALL -v1 -cpublic *snmp_agent_Ip_address* **sysName.0**

Lệnh này trả về tên được gán theo cách quản trị cho nút được quản lý. Ví dụ:

```
% snmpget -v2c -cprivate -mALL snmp_agent_Ip_address sysName.0 sysObjectID.0 ilomCtrlDateAndTime.0
SNMPv2-MIB::sysName.0 = STRING: SUNSPHOSTNAME
SNMPv2-MIB::sysObjectID.0 = OID: SUN-ILOM-SMI-MIB::sunILOMSystems
SUN-ILOM-CONTROL-MIB::ilomCtrlDateAndTime.0 = STRING: 2007-12-10,20:33:32.0
```

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

Lệnh snmpwalk

snmpwalk -mALL -v1 -cpublic *snmp_agent_Ip_address* **system**

Lệnh snmpwalk tự động thực hiện một chuỗi các yêu cầu GETNEXT được xâu chuỗi. Đó là một lệnh tiết kiệm công việc. Thay vì phải đưa ra một loạt yêu cầu snmpgetnext, một yêu cầu cho mỗi ID đối tượng hoặc nút, trong cây con, bạn có thể chỉ cần đưa ra một yêu cầu snmpwalk trên nút gốc của cây con và lệnh nhận giá trị của mọi nút trong cây con.

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

```
% snmpwalk -mALL -v1 -cpublic snmp_agent_Ip_address system
SNMPv2-MIB::sysDescr.0 = STRING: ILOM machine custom description
SNMPv2-MIB::sysObjectID.0 = OID: SUN-ILOM-SMI-MIB::sunILOMSystems
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (16439826) 1 day, 21:39:58.26
SNMPv2-MIB::sysContact.0 = STRING: set via snmp test
SNMPv2-MIB::sysName.0 = STRING: SUNSPHOSTNAME
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 72
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (14) 0:00:00.14
SNMPv2-MIB::sysORID.1 = OID: IF-MIB::ifMIB
SNMPv2-MIB::sysORID.2 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.3 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.4 = OID: RFC1213-MIB::ip
SNMPv2-MIB::sysORID.5 = OID: UDP-MIB::udpMIB
SNMPv2-MIB::sysORID.6 = OID: SNMP-VIEW-BASED-ACM-MIB::vacmBasicGroup
SNMPv2-MIB::sysORID.7 = OID: SNMP-FRAMEWORK-MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.8 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.9 = OID: SNMP-USER-BASED-SM-MIB::usmMIBCompliance
SNMPv2-MIB::sysORDescr.1 = STRING: The MIB module to describe generic objects for network interface sub-layers
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB module for SNMPv2 entities
SNMPv2-MIB::sysORDescr.3 = STRING: The MIB module for managing TCP implementations
SNMPv2-MIB::sysORDescr.4 = STRING: The MIB module for managing IP and ICMP implementations
SNMPv2-MIB::sysORDescr.5 = STRING: The MIB module for managing UDP implementations
SNMPv2-MIB::sysORDescr.6 = STRING: View-based Access Control Model for SNMP.
SNMPv2-MIB::sysORDescr.7 = STRING: The SNMP Management Architecture MIB.
SNMPv2-MIB::sysORDescr.8 = STRING: The MIB for Message Processing and Dispatching.
SNMPv2-MIB::sysORDescr.9 = STRING: The management information definitions for the SNMP User-based Security Model.
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (1) 0:00:00.01
SNMPv2-MIB::sysORUpTime.2 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.3 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.4 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.5 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.6 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.7 = Timeticks: (14) 0:00:00.14
SNMPv2-MIB::sysORUpTime.8 = Timeticks: (14) 0:00:00.14
SNMPv2-MIB::sysORUpTime.9 = Timeticks: (14) 0:00:00.14
```

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

Lệnh snmpbulkwalk

Lệnh snmpbulkwalk sử dụng tính năng giao thức GETBULK SNMP để truy vấn toàn bộ cây thông tin về một thực thể mạng. Lệnh này có thể đóng gói nhiều đối tượng hơn vào các gói bằng cách chỉ định "bộ lặp". Kết quả là lệnh snmpbulkwalk nhanh hơn lệnh snmpwalk.

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

Ví dụ lệnh snmpbulkwalk

```
% date
Fri Dec 14 12:21:44 EST 2007
% snmpwalk -mALL -v2c -cprivate snmp_agent_Ip_address entPhysicalTable>time3
% date
Fri Dec 14 12:21:53 EST 2007
```

Here is example of an `snmpbulkwalk` command performing the same operation. Notice that the `snmpbulkwalk`

```
% date
Fri Dec 14 12:40:57 EST 2007
% snmpbulkwalk -mALL -v2c -cprivate snmp_agent_Ip_address entPhysicalTable>time7
% date
Fri Dec 14 12:41:03 EST 2007
```

3.2. Lập trình Trap Receiver

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.

3.2. Lập trình Trap Receiver

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.

Các bản tin Trap Receiver

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.

Các bản tin Trap Receiver

+ 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.

Các bản tin Trap Receiver

+ 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.

Ví dụ chương trình Trap Receiver

Visual C++ project

https://www.activexperts.com/network-component/howto/snmptrcv/vc/

HTML Form SNMP Trap Receiver Sample Source Code

https://www.activexperts.com/network-component/howto/snmptrcv/html/

Simple Python based SNMP receiver

https://github.com/gabrielmarques22/simple-python-snmp-trap-receiver

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

Ví dụ chương trình Trap Receiver (code Python)

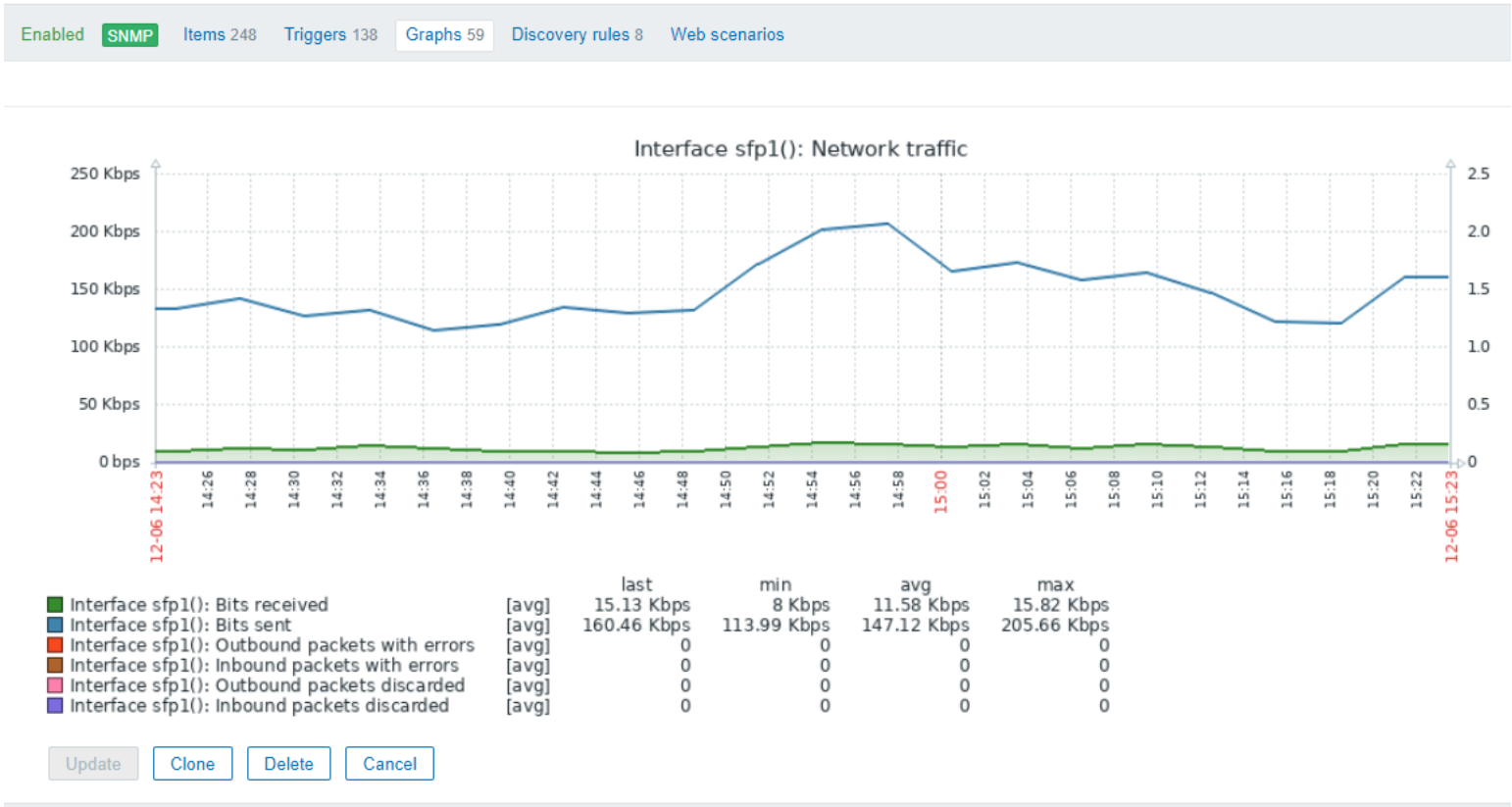
SNMPTrapReceiver.py - Notepad

File Edit Format View Help

```
#python snmp trap receiver
from pysnmp.entity import engine, config
from pysnmp.carrier.asyncore.dgram import udp
from pysnmp.entity.rfc3413 import ntfrcv
import logging

snmpEngine = engine.SnmpEngine()

TrapAgentAddress='127.0.0.1'; #Trap listerner address
Port=163;  #trap listerner port

logging.basicConfig(filename='received_traps.log', filemode='w', format='%(asctime)s - %(message)s', level=logging.INFO)

logging.info("Agent is listening SNMP Trap on "+TrapAgentAddress+" , Port : " +str(Port))
logging.info('--------------------------------------------------------------------------------')

print("Agent is listening SNMP Trap on "+TrapAgentAddress+" , Port : " +str(Port));

config.addTransport(
    snmpEngine,
    udp.domainName + (1,),
    udp.UdpTransport().openServerMode((TrapAgentAddress, Port))
)

#Configure community here
config.addV1System(snmpEngine, 'my-area', 'public')

def cbFun(snmpEngine, stateReference, contextEngineId, contextName,
          varBinds, cbCtx):
    print("Received new Trap message");
    logging.info("Received new Trap message")
    for name, val in varBinds:
        logging.info('%s = %s' % (name.prettyPrint(), val.prettyPrint()))
        print('%s = %s' % (name.prettyPrint(), val.prettyPrint()))

    logging.info("==== End of Incoming Trap ====")
ntfrcv.NotificationReceiver(snmpEngine, cbFun)

snmpEngine.transportDispatcher.jobStarted(1)

try:
    snmpEngine.transportDispatcher.runDispatcher()
except:
    snmpEngine.transportDispatcher.closeDispatcher()
    raise
```

3.2. Lập trình SNMP Traffic Monitor

Traffic Monitor rất cần thiết cho người giám sát mạng. Thông qua Traffic Monitor người giám sát mạng có thể biết được hiệu năng của thiết bị.



SNMP Traffic Monitor cho phép:

- Lấy được các thông tin mô tả thiết bị (nhóm mib-2.system).
- Lấy danh sách các interface và cho phép người dùng chọn interface để giám sát.
- Cho phép chọn các chu kỳ lấy mẫu khác nhau.
- Vẽ lưu lượng ra và xuất ra biểu đồ.
- .....

Ví dụ:

- *Để lấy thông tin về hệ thống (tên, mô tả, thời gian hoạt động, ...) ta lấy tất cả OID nằm dưới:*

.iso.org.dod.internet.mgmt.mib-2.system (.1.3.6.1.2.1.1).

- *Để lấy thông tin về tốc độ lưu lượng của interface:*

Ta phải lấy tổng số byte mà interface đã nhận tại OID .iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifInOctets (.1.3.6.1.2.1.2.2.1.10) và tổng số byte đã truyền tại ifOutOctets (.1.3.6.1.2.1.2.2.1.16)

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

Ví dụ code:

zbx-draytek.xml - Notepad

File Edit Format View Help

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>4.0</version>
    <date>2019-12-12T10:27:24Z</date>
    <groups>
        <group>
            <name>Templates/Modules</name>
        </group>
    </groups>
    <templates>
        <template>
            <template>zbx-draytek</template>
            <name>zbx-draytek</name>
            <description>Zabbix template for monitoring Draytek router/firewall</description>
            <groups>
                <group>
                    <name>Templates/Modules</name>
                </group>
            </groups>
            <applications>
                <application>
                    <name>CPU</name>
                </application>
                <application>
                    <name>General</name>
                </application>
                <application>
                    <name>Internal items</name>
                </application>
                <application>
                    <name>Memory</name>
                </application>
                <application>
                    <name>Network interfaces</name>
                </application>
                <application>
                    <name>Status</name>
```

CHƯƠNG 3. LẬP TRÌNH ỨNG DỤNG SNMP

Ví dụ code DHCP:

Item	Description
Feature name	Name of the module to which a trap belongs.
Trap number	Number of traps.
Trap name	Name of a trap. Traps on the DHCP module include: • hwNomatchSnpBindTblDhcpPktAlarm: The device sends a Huawei proprietary trap when the number of DHCP Request packets that do not match DHCP snooping binding entries and are discarded on an interface exceeds the threshold. • hwDhcpSnpChaddrAlarm: The device sends a Huawei proprietary trap when the number of discarded DHCP packets whose CHADDR field differs from the source MAC address exceeds the threshold. • hwDhcpV6PktPrefixAlarm: The device sends a Huawei proprietary trap message when the DHCPv6 prefix length exceeds the threshold. • hwUntrustedReplyPktAlarm: The device sends a Huawei proprietary trap message when the number of DHCP Reply packets discarded on an untrusted interface exceeds the threshold. • hwDhcpPktRateAlarm: The device sends a Huawei proprietary trap message when the number of discarded DHCP packets whose rate exceeds the limit exceeds the threshold. • hwSnpUserNumberAlarmIf: The device sends a Huawei proprietary trap message when the number of DHCP access users on an interface exceeds the upper threshold. • hwSnpUserNumberAlarmIfResume: The device sends a Huawei proprietary trap message when the number of DHCP access users on an interface falls below the lower threshold. • hwSnpUserNumberAlarmVlan: The device sends a Huawei proprietary trap message when the number of DHCP access users in a VLAN exceeds the upper threshold. • hwSnpUserNumberAlarmVlanResume: The device sends a Huawei proprietary trap message when the number of DHCP access users in a VLAN falls below the lower threshold. • hwSnpUserNumberAlarmGlobal: The device sends a Huawei proprietary trap message when the number of DHCP access users on the device exceeds the upper threshold. • hwSnpUserNumberAlarmGlobalResume: The device sends a Huawei proprietary trap message when the number of DHCP access users on the device falls below the lower threshold. • hwNdSnpUserNumberAlarmIf: The device sends a Huawei proprietary trap message when the

Ví dụ code DHCP:

	• hwNdSnpUserNumberAlarmIf: The device sends a Huawei proprietary trap message when the number of ND access users on an interface exceeds the upper threshold. • hwNdSnpUserNumberAlarmIfResume: The device sends a Huawei proprietary trap message when the number of ND access users on an interface falls below the lower threshold. • hwNdSnpUserNumberAlarmGlobal: The device sends a Huawei proprietary trap message when the number of ND access users on the device exceeds the upper threshold. • hwNdSnpUserNumberAlarmGlobalResume: The device sends a Huawei proprietary trap message when the number of ND access users on the device falls below the lower threshold.
Default switch status	Default status of the trap function: • on: The trap function is enabled by default. • off: The trap function is disabled by default.
Current switch status	Trap status: • on: The trap is enabled. • off: The trap is disabled.

CÁC KHÁI NIỆM CƠ BẢN VỀ QUẢN TRỊ MẠNG

Q & A