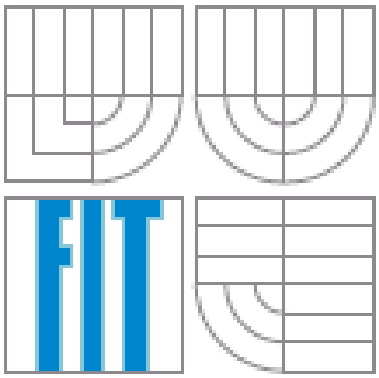

Documentation of project for ISA

Simple SNMP Manager

20th October
2016

Autor: Miroslav Cibulka, xcibul10@stud.fit.vutbr.cz
Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Content

SNMP

What is it?

Simple network management protocol is standard protocol for managing devices on IP networks. Typicaly supports devices like printers, workstations, modem racks and many more. It operates in the Application Layer of Internet Protocol Suite.

Packet

For this project I decided to use SNMPv1, because it is simple and satisfactory for our cause. Header contains version (which is 1) and community string, that is sort of authentification transmitted as a plain text. Then there is PDU (Protocol Data Unit) field, which contains whole data information needed by agent to response. It contains type of message like *GetRequest*, *GetNextRequest*, *GetResponse* and other uninteresting types. There are also data bindings in which we define name of object, we want to control or check out.

Format

Every field in SNMP message has same syntax: type of field, length of data and then data itself. Type determinates how data are saved in the message. Type of data can be signed or unsigned integer, octet string and so on. Packet is sequence, version is integer, community string is octet string and PDU is sequence.

PDU is field that contains data we send to agent such as object name and value. It contains type, request id, error status, error index and variable bindings. Every field has same format as it is in SNMP message. Type is integer, request id is integer, error status is integer, error index is integer and variable bindings is sequence. Type determines, what we want to do, for instance if we send packet signed as get request we will get in return information about object we named in variable bindings. In Error status is written what error occurred in message, such as “No such name” or “Read only”. Error index points to bindings where the error appeard.

Information about objects are in variable bindings. This contains sequence of object binds, which is sequence of object name and value. If we are sending get request or get next request, object value is null typed with no data, so we retrieve response with this field filled.

Object bind

This is typed as sequence, that contains object name and value. Both are coded in certain type. Type of object name is object identifier (for example “1.3.6.1.2.1.2.2.1.1”). First two numbers represents one byte in name. There

is a problem with other numbers. One byte is integer in range 0-255 so bigger numbers will be trucated. For that reason number are coded in 7 bits and first bit show us if number is bigger than 255 so next byte belogs to that number.

Values may have different types. For example *ifIndex* is number so type is integer, *ifDescr* is name of interface so it is coded in octet string and so on. There is several exceptions where this does not work. *ifPhysAddress* has type octet string, but octets does not represents characters, but numbers of physical address.

SNMP Manager

Manager is a program that communicates with certain device, known as agent. We can make requests for a lot of different useful informations about device. In this project we are interested in interfaces only.

What are interfaces?

Every device has inteface with which communicate in network. It has a properties like *ifNumber*, *ifIndex*, *ifDescr* and so on, that define base information such as speed, name, type and a lot of others.

How do we get them?

We don't know how many interfaces are in device, but we know what we want from them. To find out the number of devices we can simple send request for this and get the number. Easier but a lot slower solution is do it like it they do it in snmpwalk program. We send *GetNextRequest* for object before object we want and in response we get information about object we want. Looping this until unrelaited object will come, we get all information about all interfaces. It is simple because we don't have to know anything about these objects.

Decoding values

Because of exceptions mentioned before (*ifPhysAddress*) this is not a simple task. All fields in message has type with same syntax except object values. So it is wise to decode them one by one. There are five base types and these are unsigned integer (*Counter32*, *Gauge32*), signed integer (*Integer*, *Timeticks*), octet string of numbers (*ifPhysAddress*), object identifier (*ifSpecific*) and octet string of ASCII characters without zero at the end (*ifDescr*). There are slight differencies between *Counter32* and *Gauge32*, but for our purpose this is more than enough. The same with *Integer* and *Timeticks*.

Intervals

The purpose of this project is to checking out changes of interfaces in loop in the certain intervals. For that purpose I use sleep function after sending

request to agent. But there is a problem. Sending takes so much time, that if we set interval to one second, response will come for instance in 1.50 sec. So there is a stopwatch to tell the time difference and to correct interval between requests.

Time stamps

Each of response to request is printed to the screen after they are received and formated. Time stamp is printed before them. Reason why there are stamps is that we then see when and what changed in interfaces.

Implementation

Every field in SNMP packet have their own class. They inherits from Bitmap interface and ability to serialize into binary. These functions are used in Manager to create packet when receiving or send serialized packet.

Used applications

It is good to see how real packet looks like and compare them with packet send by our manager.

Snmpwalk is great tool for sending and receiving SNMP packets in different versions. To scan outgoing and incoming packets is Wireshark the first choice. We can filter packets and check SNMP communication between agent and manager. So with wireshark and snmpwalk we can compare packets with ours.