SNMP-based network-management packages up and running with very little effort.

Yousra Abdul-Sahib
Demonstrator at Computer unit,
Ibin Rushd College,
University of Baghdad

Assma Hasien Alwan
Demonstrator at Computer Unit,
Ibin Rushd College
University of Baghdad

Alla Flalah Hassen
Programmer at Computer Unit,
Ibin Rushd College,
University of Baghdad

Abstract

The Simple Network Management Protocol (SNMP) is an Internet-standard protocol for managing devices on IP networks. Many kinds of devices support SNMP, including routers, switches, servers, workstations, printers, modem racks, and Uninterruptible Power Supplies (UPSs). The ways you can use SNMP range from the mundane to the exotic: it's fairly simple to use SNMP to monitor the health of routers, servers, and other pieces of network hardware, but it can also use it to control

network devices and even send pages or take other automatic action if problems arise.

This paper is aimed toward system administrators who would like to begin using SNMP to manage their servers or routers and but who lack the knowledge or understanding to do so. We try to give a basic understanding of what SNMP is and how it works; beyond that, we show how to put SNMP into practice, using a number of widely available tools.

1.Introduction

In today's, complex network of routers, switches, and servers, it can seem like a daunting task to manage all the devices on the network and make sure they're not only up and running but performing optimally. This is where the ***Simple Network Management Protocol*** (SNMP) can help[1]. The core of SNMP is a simple set of operations as follows:

1. That gives administrators the ability to change the state of some SNMP-based device. For example, it can use SNMP to shut down an interface on your router or check the speed at which your Ethernet interface is operating. SNMP can even monitor the temperature of the switch and give a warn when it is too high.

2.SNMP usually is associated with managing routers, but it's important to understand that it can be used to manage many types of devices. While SNMP's predecessor, the *Simple*

Gateway Management Protocol (SGMP), was developed to manage Internet routers, SNMP can be used to manage Unix systems, Windows systems, printers, modem racks, power supplies, and more. Any device running software that allows the retrieval of SNMP information can be managed[2]. This includes not only physical devices but also software, such as web servers and databases.

3.Another aspect of network management is *network* monitoring; that is, monitoring an entire network as opposed to individual routers, hosts, and other devices. *Remote Network Monitoring* (RMON) was developed to help us understand how the network itself is functioning, as well as how individual devices on the network are affecting the network as a whole. It can be used to monitor not only LAN traffic, but WAN interfaces as well [3].

2. Network-Management Software (NMS)

Many SNMP software packages are available, ranging from programming libraries that let you build your own utilities to expensive, complete network-management platforms[4]. Management software falls into five categories: SNMP Agents, NMS Suites, Element Managers (vendor-specific management), Trend-Analysis and Software Supporting. Unfortunately, deciding what you need isn't as simple as picking one program from each category [5]. If you have a small network and are interested in building your own tools, you probably don't need a complex NMS suite. Whether or not you need trend-analysis software depends, obviously, on if you're interested in analyzing trends in your network usage. The products available depend in part on the platforms in which you're interested. The minimum you can get by with is an SNMP agent on a device and some software that can retrieve a value from that device (using an SNMP *get*)[6]. Although this is minimal, it's enough to start working, and you can get the software for free.

3. Network Management Software (NMS) Architectures

A better architecture is to use private links to perform all your network-management functions. Figure.1 shows how the distributed NMS architecture can be extended to make use of such links [7].

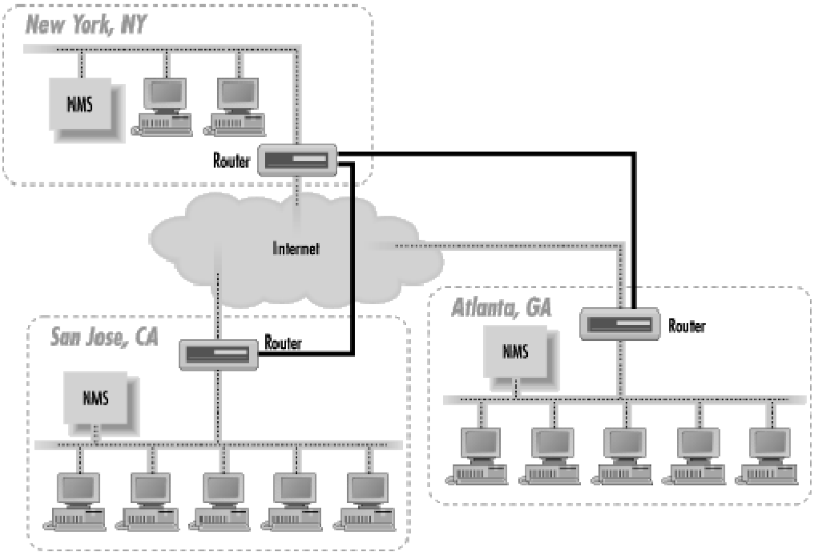

Figure.1 Using private links for network management

Let's say that New York's router is the core router for the network. We establish private links between San Jose and New York, and between New York and Atlanta. This means that San Jose will not only be able to reach New York, but it will also be able to reach Atlanta via New York. Atlanta will use New York to reach San Jose, too. The private links (denoted by thicker router-to-router connections) are primarily devoted to management traffic, though we could put them to other uses. Using private links has the added benefit that our community strings are never sent out over the Internet. The use of private network links for network management works equally well with the single NMS architecture, too. Of course, if your corporate network consists entirely of private links and your Internet connections are devoted to external traffic only, using private links for your management traffic is the proverbial "no-brainer"[8].

4. Network Node Manager (NNM)

Network Node Manager (NNM) is a licensed software product. The package includes a feature called "Instant-On" that allows you to use the product for a limited time (60 days) while you are waiting for your real license to arrive. During this period, you are restricted to a 250-managed-node license, but the product's capabilities aren't limited in any other way [8]. When you install the product, the Instant-On license is enabled by default. One of the components of the network node manger is filter.

5. Filters

It may not want to poll or manage users' PCs, particularly if you have many users and a limited license. It may be worthwhile to ignore these user devices to open more slots for managing servers, routers, switches, and other more important devices.

The wrong way can generate huge amounts of management traffic. This happens when people or programs use default polling intervals that are too fast for the network or the devices on the network to handle. For example, a management system might poll every node in your 10.1.0.0 network -- conceivably thousands of them -- every two minutes. Filtering saves the trouble of having to pick through a lot of useless nodes and reduces the load on your network [2].

Net monitoring has a filtering mechanism that allows controlling precisely which devices you manage. It lets filter out unwanted devices, cleans up maps, and can reduce the amount of management traffic on network.

Using a filter allowing keeping the critical nodes on its network in view. The last thing which wants is to receive notification each time a user turns off his PC when he leaves for the night [9].

Defining four filters: `ALL IP Routers`, `Sinatra Users`, `Markel Users`, and `Dial Access`. The first filter says to discover nodes that have field value `are Router`. Open View can set the object attribute for a managed device to values such as `is Router`, `is Hub`, `is Node`, etc. These attributes can be used in Filter expressions to make it easier to filter on groups of managed objects, as opposed to IP address ranges[2].

The next two filters specify IP address ranges. The `Sinatra Users` filter is the more complex of the two. In it specify three IP address ranges, each separated by logical OR symbols. The first range `(("IP Address" ~ 199.127.6.50-254))` says that if the IP address is in the range 199.127.6.50-199.127.6.254, then filter it and ignore it. If it's not in this range, the filter looks at the next range to see if it's in that one. If it's not, the filter looks at the final IP range. If the IP address isn't in any of the three ranges, the filter allows it to be discovered and subsequently managed by NNM [10].

The final filter, `Dial Access`, allows us to exclude all systems that have a hostname listed in the `dialup users` set, which was defined at the beginning of the file.

Filters also help network management by letting you exclude Dynamic Host Configuration Protocol (DHCP) users from network discovery and polling. DHCP and Bootstrap Protocol (BOOTP) are used in many environments to manage large IP address pools. While these protocols are useful, they can make network management a nightmare, since it's often hard to figure out what's going on when addresses are being assigned, deal located, and recycled. All servers and printers have hard coded IP addresses. With our setup, specify all the DHCP clients and then state that we want everything *but* these clients in our discovery and maps [11].

The default filter file, which is located in is broken up into three sections:

- Sets
- Filters
- Filter Expressions

Sets allowing placing individual nodes into a group. It can then use these groups or any combination of IP addresses to specify your Filters, which are also grouped by name. Then can take all of these groupings and combine them into Filter Expressions.

6. Filters Expressions

The next section, Filter Expressions, allows us to combine the filters have previously defined with additional logic. It can use a Filter Expression anywhere you would use a Filter. Think of it like this: creating complex expressions using Filters, which in turn can use Sets in the `contents` parts of their expressions [12]. It can then use Filter Expressions to create simpler yet more robust expressions. In our case, taking all the filters from above and place them into a Filter Expression called `Users`.

Since we want our NNM map to contain no found user devices, we then define a group called `Not Found` and tell it to ignore all user-type devices with the command `Users`. As you can see, Filter Expressions can also aid in making things more readable [9].

The following example which is written in language C. It should get most users up and running with some pretty good filtering.

```
//Begin of my program filters
Sets {

   dialupusers "DialUp Users" { "dialup120", "
dialup121","dialup122","dialup123", }

}

Filters {

   ALL IP Routers "All IP Routers" {is Router}

   Sinatra Users "All Users in the Sinatra Plant" { \
      ("IP Address" ~ 199.127.5.50-254) || \
      ("IP Address" ~ 199.127.6.50-254) || \
        ("IP Address" ~ 199.127.7.50-254) || \
        ("IP Address" ~ 199.127.8.50-254)  }

    Markel Users " Users in the Markel Plant" { \
        ("IP Address" ~ 172.247.63.17-42) }

    Dial Access "All Dial Access Users" { "IP Hostname" in
dialup users }
}

Filter Expressions
{
    Users " Users" { SinatraUsers || MarkelUsers || DialAccess }

    Not Found "No Users” {! Users }
}
```

7. Conclusions

When faced with most network problems, it's nice to have some kind of historical record to give you an idea of when things started going wrong. This allows you to go back and review what happened before a problem appeared, and possibly prevent it from recurring. If you want to be proactive about diagnosing problems before they appear, it is essential to know what "SNMP" means for your network -- you need a set of baseline statistics that show you how your network normally behaves. While many of the bigger packages do some trend reporting, they can be clunky and hard to use. They might not even provide you with the kind of information you need. Once you see what a NMS can do, you will see why it might be worth the time, energy, and money to integrate one into your network-monitoring scheme.

Now that you have picked out some software to use in your environment, it's time to talk about installing and running it. In this paper we will look at a few NMS packages in detail. While we listed several packages in we will search into only a few packages here, and we'll use these packages in example as filter. This example should allow you to allows you to keep the critical nodes on your network in view. It allows you to poll the devices you care about and ignore the devices you don't care about.

8.References

1. Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards",   April 1988.
2. Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", TWG, August 1988.
3. McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", TWG, August 1988.
4. Cerf, V., "Report of the Second Ad Hoc Network Management

Review Group", IAB, August 1989.

5. Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", Performance Systems International and Hughes LAN Systems, May 1990.
6. McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets", Hughes LAN Systems and Performance Systems International, May 1990.
7. Case, J., M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol", Internet Engineering Task Force working note, Network Information Center, SRI International, Menlo Park, California, March 1988.

8. Davin, J., J. Case, M. Fedor, and M. Schoffstall, "A Simple Gateway Monitoring Protocol", Proteon, University of Tennessee at Knoxville,

Cornell University, and Rensselaer PolytechnicInstitute, November 1987.

9. Information processing systems-Open Systems Interconnection, "Specification of Abstract Syntax Notation One (ASN.1)", International Organization for Standardization, International Standard 8824, December 1987.

10. Information processing systems - Open Systems Interconnection, "Specification of Basic Encoding Rules for Abstract Notation One (ASN.1)", International

11.Comer, Douglas E. *Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture*. Third Edition. Englewood Cliffs, NJ: Prentice Hall, 1995.

12.Nemeth, Evi, Garth Snyder, Scott Seebass, and Trent R. Hein. *Unix System Administration Handbook*. Second Edition. Englewood Cliffs, NJ: Prentice Hall, 1995.