INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Monitoring Network through SNMP-based System

Jianqing Liu*, Rongkai Lu

School of Information Technology Engineering,
Tianjin University of Technology and Education, Tianjin, China

* *Corresponding author's Email: foreverljq@gmail.com*

Abstract: SNMP-based system for monitoring network is equipped with network management, monitoring system and statistic analysis of the network in it. It not only contains the function like network information capture that normal network devices have, but also has the ability to extend such applications as CPU usage rate, traffic flow information of interfaces and memory usage rate through plug-in mechanism. Simplifying the basic network management tasks by centering on auto-topology mechanism and making a tradeoff between usability and extensibility, the auto-topology control which is implemented and designed independently could support almost networking devices with better flexibility. Completely supporting to capture TrapV1, TrapV2 and partial TrapV3 (limited by protocol) with embedded database to facilitate backend storage, the software has become a green-software except depending on .NET Framework 4.

Keywords: SNMP; Automatic Topology; Weight Priority.

1. Introduction

SNMP is widely used for the monitoring, management and controlling of the network devices. Now in its third release, SNMP has become the de facto standard for network management since its development in 1987 [1]. Thus, it is a full-fledged network management protocol [2]. As we know, different network devices need to be supported by different programs; for example, Literature [3, 4, 5] focus on single function like traffic monitoring or topology display regardless of the scalability and generality of the system based on SNMP; so the SNMP-oriented matured framework of network management system is few and far between. Cisco Company is at the leading edge of supporting SNMP [6]. It not only provides fully support to the definition of MIB-2 in RFC1213, but also supports RMON of RFC2819 and RFC2021 embraced within some devices. Therefore, it is of significance to develop extensible SNMP-based network framework to support the prevalent Cisco devices.

Two basic and important problems are confronted to software development when talked to programming; they are generality and scalability respectively. Generality can assure devices' basic functions to perform well in most networking environment while scalability can guarantee their special functions on the condition that the generality of software is well conducted, which makes the job designing general framework much more difficult.

How to simplify the network management tasks catches more attention. This is mainly because the operating behavior can be affected by the working way of software selected. C/S pattern or pure single-structured software no doubt has powerful graphic presentation while B/S pattern can show its merits on scalability and cross-platform. A tradeoff and efficiency alternative is to implement generality and cross-platform on the level of framework while different solution can be adopted on this presentation.

2. System Analysis

2.1 Key issues

Since the supports to networking devices furnished by SNMP are different between devices, developing some general functions attached in software, which are compatible with most of networking equipment, is challengeable. On the basis of topology as its blueprint, the common network management tasks can be fulfilled properly. Therefore, no matter what diverse the networking equipments are, and how complicated the environment supported by SNMP is, automatic network topology discovery is undoubtedly an indispensable and core function. Thus, how to extend function on common compatible equipment becomes the focus of our research.

2.1.1 Compatible with most of existing equipment

Rather than realizing compatibility, it would be better to implement the functions backed by most hardware. One solution to it is to utilize MIB-2, the functional groups defined by RFC1213, which releases the routine objects in networking administration and gains the general acceptance from most devices.

2.1.2 Obtain network topology through SNMP

There are no objects in functions relevant to SNMP which can be directly used to obtain network topology. But they can be found in some MIB which is in the possession of the private where Cisco can be taken an example. Some objects related to CDP (Cisco Discovery Protocol) from Cisco conduct such functions of getting the topology immediately. However, it poses very sticky requirement on network environment and the equipment for CDP must be used in pure network Cisco-based, although a few non-Cisco start to give support to CDP. Thus, it is not a solution for general purpose.

To keep the compatibility between network equipment, the conservative objects should be the priority to implement the core functions. And the automatic topology discovery can be deployed until we import emphasis on the MIB-2. The scheme is to find out the address translation table with the information about IP of PC among it. With these IP, the SNMP testing on PCs can be employed; consequently, the SNMP-based network is well-supported.

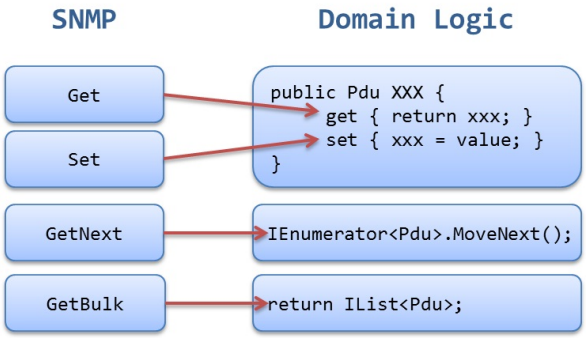

Figure 1 Domain for Transformation

2.2 Difficulty in software design

2.2.1 Layout of topology

Once the mechanism of topology is confirmed, an intractable issue, the layout of devices and devices' links, blocks the way to further the job. It is too hard to draw the topology the same as the physical structure totally by software.

During the course of developing software, the strategy pattern is adopted to abstract the layout algorithm, so we can do some substitution manually.

2.2.2 Mapping from domain model to storage model

The domain model is regarded as conceptual model of a system, which is used to describe the relationship between entities and their relations visually. For an application system based on SNMP, the domain model is its main entity. Additional extensive function, managing the network devices, is associated with such application as assets appraisal, devices accounting, and maintenance.

3. Framework Design

3.1 Logic design of domain

The main logic of Software to interact with SNMP network is dependent on the SNMP object data transmission by SNMP protocol; meanwhile SNMP objects are dependent on the relevant MIB to describe its characteristics and structure. Software needed logic is mainly concentrated on the operation of the SNMP entity, but not friendly to the program, that is, not through the smooth operation of API to make use of the software for SNMP.

So it is necessary to design domain logic to convert specific domains of SNMP into program friendly domains. Now consider the domain transformation shown in Figure 1.

```
[Serializable]
public sealed class DeviceBasicInfo:BasicModel
{
    [Single(".1.3.6.1.2.1.1.1.0")]
    [AccessControl]
    [OneTime]
    public String Description { get; set; }

    [Single(".1.3.6.1.2.1.1.3.0")]
    [AccessControl]
    [OneTime]
    public String UpTime { get; set; }
}
```

Figure 2 Model with DSL Characteristics

Hierarchy	M. I.	C. C.	I. D.
PowerManager.Application (D	100	0	0
PowerManager.Framework (D	89	192	3
PowerManager.Module\AutoT	78	45	7
PowerManager.Module\AutoT	90	3	1
PowerManager.Module\Power	82	131	3
PowerManager.Module\Power	87	204	7
PowerManager.Module\Power	81	102	6
SManage\SManage (Debug)	57	134	7
Test\SManage.TestConsole (D	98	4	1

Figure 3 Measurement Result of Code

From the figure above, SNMP operating primitive is converted into the corresponding programming concepts, so that the domain of SNMP completely turns into the programming domain. This provides the basis for the expansion of AOP programming and storage model.

3.2 DSL expressive force

DSL is a Domain-Specific Language [7], according to Martin Fowler's view; DSL is also a tool which can help users abstract a certain part from a system.

SNMP protocol itself is designed as an associate network management tool, so it is a pity that if this property is abandoned and a new administrative model is developed from the beginning. Starting from a specific language and using this feature of SNMP, AOP (aspect-oriented programming) model is made to achieve the specific features of DSL, which keeps the language and the framework itself easy to use and simplifies programming.

The benefits of using DSL features is obvious, for data with strong type could be employed in the design stage, and the introduction of caching strategy and parallel optimization in later design step can become easy by using excellent AOP framework. A relatively simple example of using AOP with SNMP features is shown in Figure 2.

Store SNMP-related data into metadata and obtain SNMP entity through reflection, which not only hides the SNMP communication at the bottom, but also gets the "shallow buffer". This is the expression force of DSL.

3.3 Scalability and flexibility

Matured framework is often evolved on the basis of the continuous evolution, and a reasonable design is the premise of this kind of smooth evolution. Software takes the strategy of the incremental evolution–every time a small iteration to design and implement the software framework–this makes framework maintainable in later, meanwhile, better performance can also be shown in scalability and flexibility.

The framework's basic extension point lies in the incremental expansion of plug-in applications with centered as topology, implementing the function of software extensions on the whole. Through the AOP model implemented and the application of the plug-in mechanisms, the software has great flexibility in the function and the maintainability for "upgrade" with no side effects available. The following Figure 3 shows measurement results of parts of code.

From the figure above, SNMP operating primitive is converted into the corresponding programming concepts, so that the domain of SNMP completely turns into the programming domain. This provides the basis for the expansion of AOP programming and storage model

The "maintainability index" 57 is the value of SManage project, the most top-level application, according to Robert C. Martin's point of view, the top-level application and framework of the maintainability index is often pyramid-shaped distribution. The maintainability index of core framework–Framework and Topology Control are respectively 89 and 87, which basically reaches the maintenance needs later.

3.4 Storage model

Storage model can provide the persistence mechanism for the relevant extension applications; meanwhile, it plays the essential role for statistic and analysis which request the comparison of history data. At the earlier stage of our design, a work model called synchronous engine was put forward, which could synchronously store the MIB-2 information of found devices into storage model in fixed time. The work model of Sync engine is illustrated in Figure 4.

However, when synchronous engine was tested, the

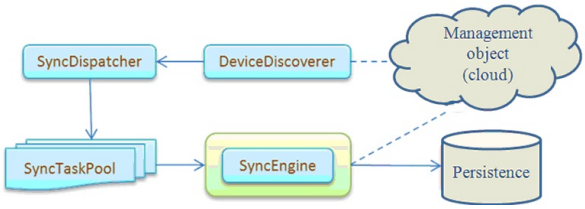

Figure 4 Working Model of Synchronous Engine

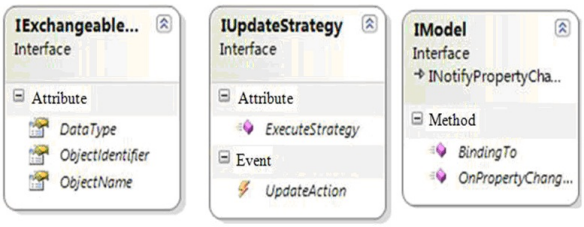

Figure 5 Public Interfaces Provided by Framework

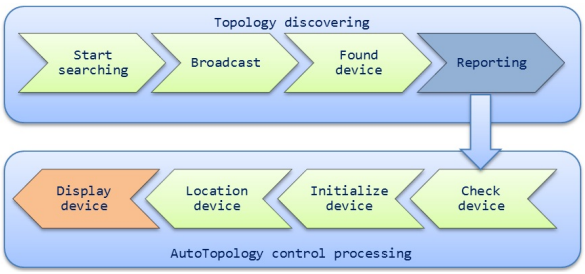

Figure 6 Process of Auto Topology Discovery

discovery from test revealed that it wrote and read storage model more frequently and the increasing amount of data was higher than before after every update. So the sync engine was discarded finally. From the angle of cost and deployment, the selectable database products like Microsoft SQL Server Express, Microsoft SQL CE, MongoDB, NoSQL, MongoDB, and NoSQL can solve the problem of persistence. Combined with the performance, Microsoft SQL CE is an alternative.

3.5 Testability

Agile development declaration says:

The individual and interaction is more important than process and tools;

Working software is more important than detailed documentation;

Corporation with Clients is more important than contract negotiations;

Prompt response to changes is more important than following a plan.

The rapid response to changes reflects the software's capabilities in scalability, maintainability, robustness and so on. And the key technology of quick responding to changes is the continuous integration and TDD (Test Driven Development).

In response to rapid changes in software, as well as taking into account the development cycle which does not allow much time to do the test and consider other factors, the software following up the part of the principle of "agile development" puts forward three principles to help software sustainable evolution: (1) Using Microsoft Visual Studio Team Foundation Server for source code version management and continuous integration; (2) give priority to the use of "ink test", the core mechanism of "white box"; (3) add a small incremental iterative function.

Software easy to test is rooted in the clear responsibilities of "class". Through continuous reconstruction [11], the software refines the core mechanism of the abstraction level; for example, Figure 5 shows the AOP abstracts model part of the interface, which makes it easier to test.

4. Detailed Design

4.1 Automatic topology control

The automatic topology Control is the core of SNMP-based system which is the starting point of other extensions. This control also provides functions such as adjusting topology, storing topology, loading topology and storing images. More details about implementing the function of automatic topology control are involved.

4.1.1 Operating mode

Obtaining topological graph is a high-cost and time-consuming process. Asynchronous discovery avoiding the congestion of UI threads is the best choice. In the mechanism of topological graph discovery, a special "hook" is added in. This is the observer pattern [8, 9, 10] which is normally employed in design field. When a station is found, mechanism will notify the subscriber of this event. The station will be shown as an image corresponding to it in the software picture.

When topological mechanism finds a device, it will report this event. The automatic topology control subscribing this event will receive the report that stands for a device. Then automatic topology control will check the device which this report stands for. Initialization will be started until the checkup is finished. Initialization will complete a series of job to the corresponding devices. Context Menu extension Mechanism is added in at this time. After initialization, the algorithm of locating device will be conducted to locate devices. Finally, the device discovered is added

into the visual panel. Thus, the addition of device is finished.

4.1.2 Double buffers

The links among devices of topology graph are drawn by GDI +. An obvious problem is that topological graph will blink when we adjust topological graph manually. This problem has nothing to do with .NET platform, but it is caused by the efficiency of the GDI+ tool. Double buffers can reduce the frequency of drawing in order to speed up the drawing rate, which can solve the drawing blink.

If DirectX is used to draw the topology, another difficulty, interoperation of COMs, is brought in. Studying API related to Windows no doubt increases the cost of technique with a bit higher efficiency than GDI+.

Selecting GDI+ leads to the efficiency of itself. The reason of using double buffers is that speeding up the rate of drawing and reducing the blink may slow down the drawing frequency.

A buffer container should be ready for the usage of double buffers. This software adopts the default implementation provided by .NET. For instance, the following code will start double buffers.

$$var\ buffered Graphics = Buffered GraphicsManager .Current.Allocate(graphic, DisplayRectangle);$$

Then, the buffer can be used to draw pictures.

$$buffered Graphics.Graphics.DrawLine(Pens.Black, p1,\ p2);$$

Finally, pictures in buffers are drawn into visual panel after all the drawing is done.

Double buffers decrease the blink but lead to another problem–transparency of the control, which brings about bad experience to users.

4.1.3 Smoothing mode

Double buffers can eliminate the drawing blink, but there is still one more problem; that is unbeautiful graph interface. Of course, a line with aliasing is ugly. Smoothing mode can be employed to clear the aliasing, which is called anti-aliasing.

In terms of drawing, there is no difference between the utilization of buffer container and GDI +. Thus, the setup of smoothing model is as same. Drawing curve using smoothing model needs to set the Smoothing Mode property of Graphics before drawing.

Usually it is enough for system to only set the property Smoothing Mode. The other properties appear in the occasion where there are many curves. However, the more properties are set, the more cost the diagram drawn by GDI+ will take.

4.1.4 Synchronization context

Event mechanism supporting asynchronous discovery solves the problem of real-time response. But another serious problem occurs. The thread used to discover topology is not the same one as UI thread. If they are the same, the asynchronous model cannot be implemented.

Topology mechanism will notify the subscriber once it finds device(s), and at that time the functions, which have registered the events, are evoked. But the evoking thread is still the thread itself discovered by topology, not the one of UI threads. If there is no special mechanism to tackle it, the exception will be thrown out.

An unofficial solution to it is to explicitly shut down the checkup of thread in the constructors of UI achieving this effectiveness by setting the property of Control.CheckForIllegalCrossThreadCallsas false. However, it is not safe to operate UI through cross-threads for it can invoke deadlock. Therefore, a mechanism transferring UI operation to its own threads is in the urgent need.

.NET platform provide a mechanism to guarantee the synchronization, which is called Synchronization Context [12]. Synchronization Context has many versions, such as Windows Forms Synchronization Context fit for Win FormDispatcher Synchronization Context used for the technique of WPF and Silverlight, ASP.NET Synchronization Context for ASP.NET and Synchronization Context based on threads. All of them have common in unified abstract, though the implementation of them are different from each other.

Since UI of the system is implemented by Win Form, Windows Forms Synchronization Context is the priority.

Retile Searching implements the interface of ITopologyStrategy, so the events subscribed from Found Station, found during the topology discovery can control asynchronous present.

4.1.5 Serializability

Topology structure as a structure of software is stored in the memory of host, and its presentation is a structure of Collection¡Station¿. The main tasks of storing and loading topology are to store or restore this structure into persistence mechanism or memory. The mapping from data structure to persistence mechanism

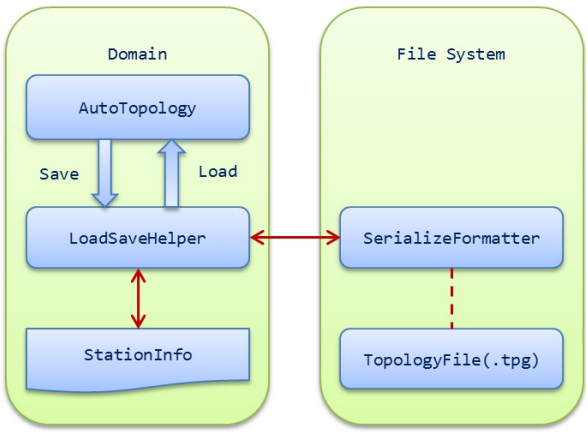

Figure 7 Working Process of Serialization

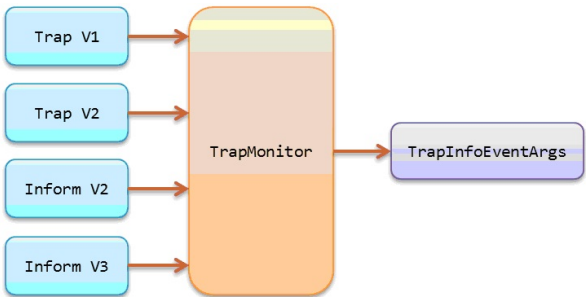

Figure 8 The Intermediate Layer of Trap

needs a mapping, but the programming is very time-consuming. The topology stored in form of a file is a good choice because it can be used conveniently and easily.

Take all the above into account, serialization is the right scenario and .NET provides the function of serialization. What we need to do is to mark "serializable" on the data structure that needs to be serialized. The process of serialization is illustrated as Figure 7.

Auto-topology control forwards the request of save/load to the class LoadSaveHelper, then LoadSaveHelper converts it into the structure of StationInofo instead of directly serializing it into Controls set of AuTopology. StationInofo keeps such information as IP address, coordinates, and links and so on. The real serialization is conducted in Serialize Formatter.

Serialize Formatter is classified into many types, such as binary serialization, XML serialization and open standard SOAP serialization. Since the interoperation between platforms is temporally not involved, binary serialization with higher efficiency is a priority.

Topology is saved into a file with the suffix .tpg by default under the default directory Data. When the system starts, it will check up the directory Data. If there are files with suffix .tpg, a dialogue box will appear to enquire if the last topology is loaded. If there is a topological file, the system will make a new topology according to the creating time of the files.

4.2 Device management

For every found device, SNMS can check its information and generate an exclusive hash code which identifies the device. SNMS is in charge of the following information.

1) Location information: The logical location in topological graph is not the physical location. SNMS should hold the physical location of the devices.

2) Device parameter: The statistic of device parameters can be helpful for maintaining the devices at later stage.

3) Information of equipment: Assets appraisal always involves the records of purchase, providers, and other relevant information, which is the starting point of maintenance and statistic.

4) Maintenance record: It is important for network administrator to know the running state of the networking equipment. The system provides users with this function regarding to adding or updating maintenance records.

4.3 Trap detection

There are many versions of traps. In order to be notified when events occur, the system needs a kind of mechanism which can receive trap and unify the different versions of traps. The system uses intermediate layer to act as an agent shown in Figure 8. Except multiple versions, the multiple types of traps make the software analysis a time-consuming job. SNMS could not predict the emerging traps with new definition because of its scalability. Modeling trap, abstracting its core into an expandable and configurable mode and implementing it by XML file can make SNMP suitable for different environment and easy deployment.

Filtering useful trap plays another import role, which is determined by the management property of software. Filtering mode, white list and black list, is used in filtration. White list shows the traps matching the rules in the list while the traps mismatching the rules in the list are in black list.

4.4 The application extension of context menu

As a major method of extending software, the context menu of devices expresses itself in this way. Since the extension mechanism is added into topology discovery, the extension of the context menu behaves globally. The system carries three extension functions it-

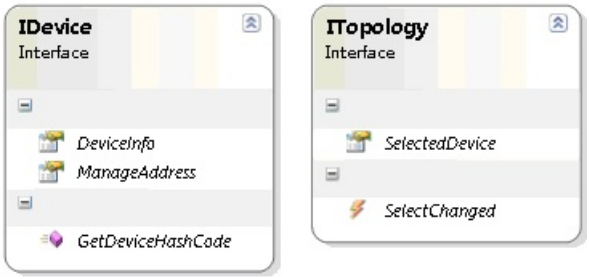

Figure 9 Definition of interface IDevice&ITopology

self used for network routine tasks, which are provided by the form of plug-in.

Every device has its unique identification to support Context Menu. Thus, the system framework offers the interface IDevice and designs the interface ITopology for assistance to topology mechanism. Figure 9 gives the definitions of mentioned two interfaces.

IDevice, as the core interface of Context Menu, is implemented by all the devices, so the plug-in can re-extend surrounding to IDevice. Compared with IDevice, ITopology interface supports the conversion between different devices.

IDevice only provides the basic function in the early software making period, and it does not offer functions like grouping and extending icon, which will be considered in later development by gradually extending more controls on this interface.

4.5 Algorithm of automatic topology

The algorithm called weight circular distribution is mainly due to introducing the weight of devices in the network. As the goal of layout is to distribute the major devices in reasonable and correct places in the screen, this kind of topology layout make the finding of devices with highest weights as its first task. If sort and locate them in the weight orderthis algorithm is comprised of four steps as follows:

The first step is to sort devices by weight. The system is routers and switches oriented. Therefore, the most sensitive information within topology is so relevant to routers. A device is regarded as a core one, which is always equipped with a wealth of routings compared with others. Though the case like above is contended to be absolute, it is the fact in most of the time. Generally speaking, the number of devices with more routings is not so much than the one with less routings. Consequently, the ones called core devices should be definitely placed on the core location, and this is what the second step will do.

Locating by weight priority is the second step. Once

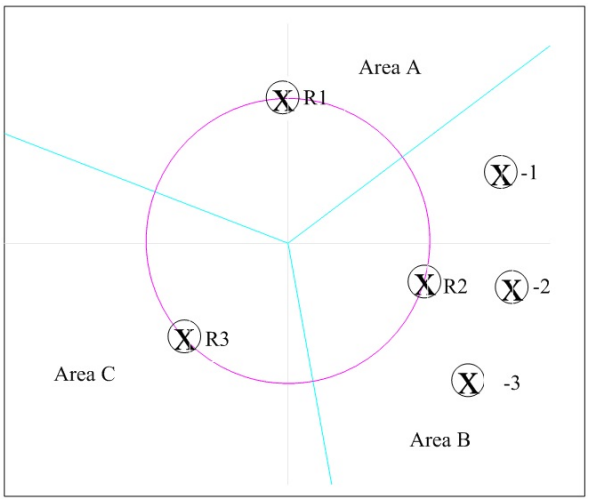

Figure 10 Distribution of satellite devices

the core devices have been settled down as well as the device group according to the weight, the first step is terminated. Then the ones in first group, the most crucial devices, averagely distribute on the circle shared the same length of radius and centered as the midpoint of the screen. To define the radius often needs some steps as follows. The more devices there are, the bigger the value of radius is. When the devices locating is done, the angle of the midpoint of the screen for the corresponding device is confirmed. The angle, thus, becomes the starting point of next step.

Here comes the third step–"satellite" devices distribution. The devices adjacent to the core devices are named and classified into satellite devices. In this algorithm the conception of aggregation is not taken into account, but the satellite devices used take the place of it. Now an example is taken of the algorithm for the distribution of satellite devices.

Assume that there are n core devices. The satellite devices for every core device can only be within the sector of 360/n. Illustrated in the figure below.

There are three core devices shown in the picture which is divided into three sectors. Herein, take $R2$ as an example. Its three satellite devices are well-distributed in the sector B according to the angle of $\angle\theta$ and the radius can be adjusted following the amount of the satellite devices. Another correcting algorithm can be like this. Observe if devices in other sectors are apparently less than the ones in current sector or not. If they are, distributing satellite devices will invade into the adjacent sectors in order to utilize the space of screen to the full. Then, the satellite devices of satellite devices distribute their topology using the same algorithm.

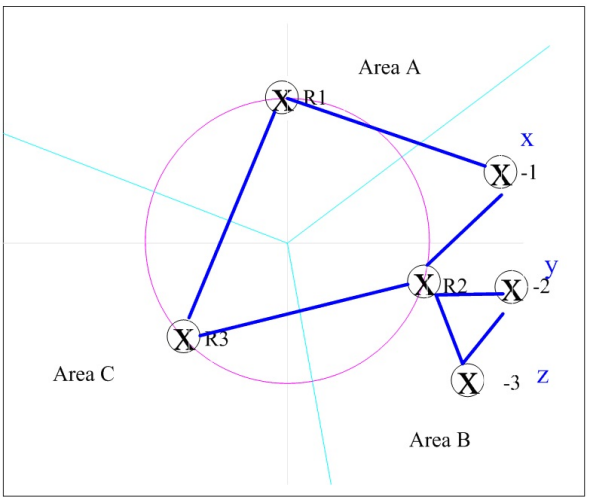

Figure 11 Drawing Links

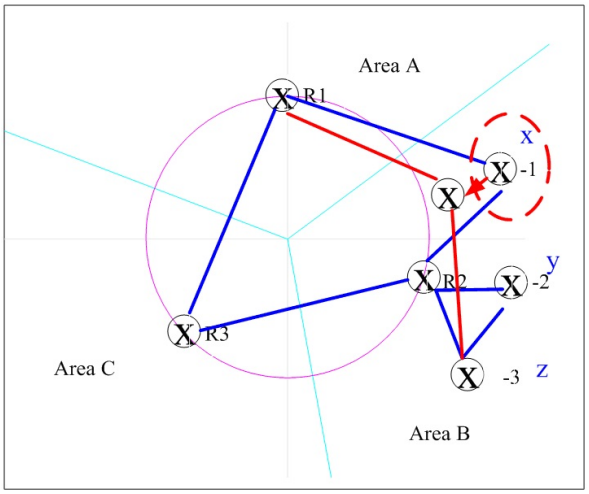

Figure 12 Adjust for topology

The fourth one is to draw the links. The intersection is permitted for links to core devices as there are almost no possibilities to be not intersected. The problem of intersection cannot become severe because of the guarantee furnished by annual distribution of devices. The links between core devices and satellite devices can be directly connected. However, if the connections happen to satellite devices, the layout of satellite devices should be slightly adjusted so that the overlaps can be avoided. See the figure below.

Currently, if there are connections between device X and device Z, the slight adjustment will be conducted in accordance with the space of the screen. Thus, the distribution of links from X to Z seems much more reasonable as shown in the figure.

The direction of devices moving follows their upper devices, of course, which is determined by the space of screen. That is, near to the upper or away from the upper.

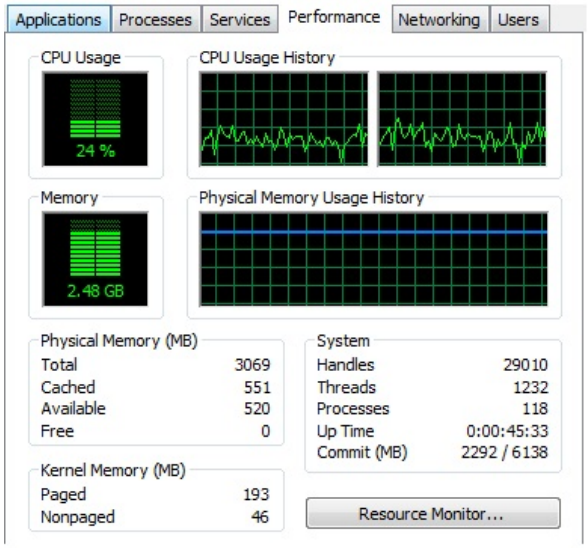

Figure 13 Simulation Testing

Another moving is longitudinal. For example, when Y is going to connect to $R3$, $R2$ is in the way obviously. The adjustment to y is neither next to $R2$ nor away from $R2$, but in the direction of coordinate axis Y. Consequently, further judgment should be made to correct the links and the best correctness is to move y down shying from $R2$.

There is still one problem needing to be solved, that is to set a value of a pixel, for instance, 50px, which can be used to adjust when the correctness should be done if the distance from current link to the device in the way is too small. If the distance is smaller than 50 pixels, adjust the location of devices, and then the topology is changed accordingly.

5. Testing and Deployment

5.1 Testing

First, simulation testing is conducted under the Cisco-simulating environment with matured simulation software. Figure 13 shows the system's performance with 10 devices in the LAN.

Second, the real environment testing is conducted. Medium-sized data exchange network of routing type is used as testing environment. It consists of 5 Cisco 7200 routers and 7 Cisco 3640 switches. After the configuration of the routing protocols, the SNMS is setup as well as the trap function. The real topology of the mentioned network is illustrated as Figure 14 before the test is done.

Using the system SMNS (SNMP-based Monitoring Networks System), the testing result is displayed as

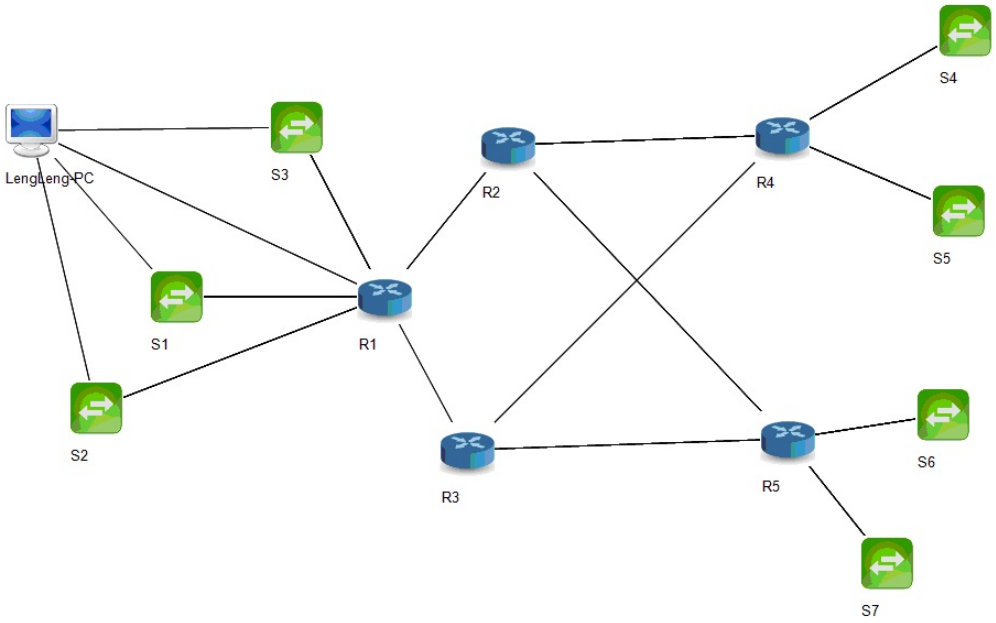

Figure 15 Testing Result

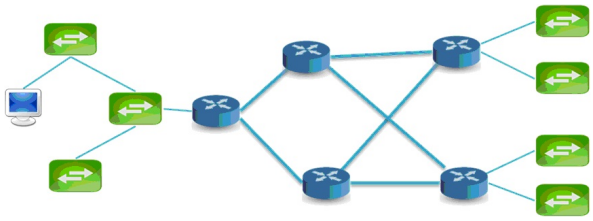

Figure 14 Topology of Network

Figure 15.

The discrepancy between original topology and testing result is caused by the switch's transparent role in the network, which influences the judgment of topology logic. But the problems can occur until the topology happens to the computers.

5.2 Deployment

1) Compile: SNMS aiming to serve network management could probably be run on X64 platform. Therefore, the guarantee that SNMS can be properly utilized on the platforms with the architecture of X64 or X86 should be taken into account. The best option for compiling the applications on different platforms is the compiling way-AnyCPU as one of the inherent attributions of .net to make compiling easy.

2) Installer: SNMS depends on the .NET Framework, so the computer without .NET Framework installed could not run SNMS properly. Packing .NET Framework together with SNMS facilitates the users. The software package should also include Windows Installer.

3) Running platform: One advantage of .NET platform is independent of running platform. .NET is a lay of abstracting between SNMS and OS. Different platforms, such as Linux, Mac OS, IOS, and Android can run on the core framework of SNMP.

6. Conclusion and Future Work

As a kind of software used in upper application based SNMP, SNMS not only implements the core mechanism of topology discovery, but also perfects the software frame, which makes it fit to develop different upper application. The ideal evolution of software is to make a basic framework based on SNMP, thus, other applications can be extended on it continuously. Due to the maturity of SNMP, this kind of frame has great potential to share the future marketing.

The design of framework is an evolutionary process. This requires that back compatibility should be considered when the increment and deletion of every API is done. The first edition of SNMS only provides conservative APIs for upper applications. With the evolution of frame, more APIs and functions will be provided so as to simplify the difficulty of developing upper applications.

The future direction to develop SNMS is to extend

its transversal functions, but not application functions. Performance and buffer are the major direction to work. For example, the introduction of parallel concept can speed up 5∼10 times in auto topology control, of course, the complexity of developing job will rise up.

Acknowledgment

This work was sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry. This work was also supported by Tianjin Research Program of Application Foundation and Cutting-Edge Technologies Grant 10JCYBJC26100.

References

[1] Jiang Guofeng, "Multiple vulnerability in SNMP," Computer, *IEEE Computer Society*, vol.35, pp.2-4, April 2002.

[2] Mingjiang Li, "SNMP simple network management protocol", Beijing: Electronic Industry Press, 2007.

[3] CHEN Shu, ZHANG Jiang-Xin, "Design of the Network Topology Display M odule in SNM P-Based GPON Network Management System", *Computer Systems & Applications*, vol.1, pp.84-88, 2011.

[4] ZHANG Tong, WU Shirong, "Research on Computer Network Traffic Monitoring System Based on SNMP," *Computer Technology and Development*, vol.21, pp.88-91, Jan. 2011.

[5] CHU Jiu-liang, "The Construction and Realization of Network Application Servers Monitoring Platform based on SNMP," *Research and Exploration in Laboratory*, vol.29, pp.65-68, Jan, 2010.

[6] Alexander Clemm, "Network Management Fundamentals", 1st ed., *Cisco Press*, 2006.

[7] Suman Pandey, Mi-Jung Choi et al. "IP Network Topology Discovery Using SNMP". Korea: Dept. Of Computer Science and Engineering.

[8] ZHANG Yi, "the essence of software design and model", Beijing: *Electronic Industry Press*, 2007.

[9] Wang Xiang, "Design Patterns-Engineering Implementation and Extension Based on the C#", Beijing: *Electronic Industry Press*, 2009.

[10] Alan Shalloway, James R. Trott, "Design Patterns Explained A New Perspective on Object-Oriented Design", Beijing, *China Electric Power Press*, 2006.

[11] Martin Fowler. "Refactoring-Improving the Design of Existing Code". *Addison Wesley*, 1999.

[12] Jeffrey Richter. "CLR via C# 3nd". *Microsoft Press*, 2010.