Expert Systems with Applications 38 (2011) 10320–10342

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An active and intelligent network management system with ontology-based and multi-agent techniques

Sheng-Yuan Yang [a,*], Yi-Yen Chang [b]

[a] Dept. of Computer and Communication Engineering, St. John's University, Taipei, 499, Sec. 4, Tam-King Rd., Tam-Shuei, Taipei County 25135, Taiwan, ROC
[b] Dept. of Electrical Engineering, St. John's University, Taipei, 499, Sec. 4, Tam-King Rd., Tam-Shuei, Taipei County 25135, Taiwan, ROC

ARTICLE INFO

Keywords:
Intelligent Agents
Graphic monitoring systems
Network management
Network flow fetching

ABSTRACT

This paper presents a system to collect information through the cooperation of intelligent agent software, in addition to providing warnings after analysis to monitor and predict some possible error indications among controlled objects in the network. This technique derived from the ontology combining Ethereal and Cacti, which store the operating information of network management perfectly into the backend database. The system could sketch the four main components of network management systems with the technique of graphic monitoring multi-agent: an Interface Agent, a Proxy Agent, a Monitoring Agent, and a Search Agent. This architecture can effectively enhance and improve the network monitoring performance to be an active and intelligent network management system. It can present related quantification figures of dynamic information through graphic network monitoring system to provide fast, convenient, and profound network solutions to the users. The experimental outcomes proved that the techniques could not only precisely recognize error alarms but also indeed reduce the recovery time to 61% of traditional processing time for network troubleshooting.

1. Introduction

Nowadays, along with popularity of application and use of network technologies, it increasingly made network be complicated and enormous. How to effectively manage various network segments and equipments, understand their problem symptoms and accordingly bring up corresponding advices with the intuitively graphic interface at the right moment so as to promote network service quality and performance has become a very important challenge in the modern network management (Lu, 2005).

In the rapidly developing era of Internet, the network environment changed from the closed one with single factory into the open heterogeneous one with lots of factories. Variously different brand and type of network equipments and software had been combined, which not only caused in oppositely raising the appearance probability of network problem but also increasingly deepened the monitor difficulty. Therefore, the network management standard in various network product environments had been driven by the international standard organization and standard organizations of countries. The network management platforms and equipments of various network layers had been produced by numerous factories, which can assist the network management staffs in effectively monitoring and managing every situations of network (Saturday, 2008). However, existent network management systems not only have different management applications of equipments and flow to make network management staffs open those management applications to monitor corresponding part of network, even more seriously in those network software are not enough for humanity and lack for flexible monitoring mechanisms, which make only network management staffs understand those monitor information during network problems were produced (Huang, 2008), which cannot be clear to general users that cannot truly satisfied real requirement of client end. Therefore, lots of businesses try to do various solutions, for examples, form analysis, obstacle warning, information security monitoring, and network performance. In general cases, the marketing products cannot be aimed at specific network service monitoring to provide completely and totally solutions.

To be aimed at the above problems, even if there are various and different solutions in the market, for examples, HP OpenView, CA Unicenter TNG, and IBM Tivoli. The software could fit in with formidable and heterogeneous network management described before but their purchase and construction fees are huge, which caused in the big cost to the business. Even though there are small and proper network managing software, for instance, WhatsUP and NetVCR. Their module architecture, need which

* Corresponding author. Tel.: +886 2 28013131x6394; fax: +886 28013131x6391.
E-mail address: ysy@mail.sju.edu.tw (S.-Y. Yang).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.02.115

functions and then purchase corresponding modules or limited their managing amount of network equipments, directly caused some persecutions to business and its network management staffs, detailed below.

(1) How to summarize the problem from a complex network structure easily, such as the continuity problem of network transferring quality.
(2) Due to the lack of integrated unity and accurate data information, it was unavailable to provide an intelligent integrated information dynamically with single platform and searching problem with sharing relativity comparatively.
(3) The heterogeneity network control could not provide a highly intelligent integrated data because of the loosed and asynchronous product information.

This study focuses on how to effectively integrate different networking devices under various enterprise environmental demands to develop a network monitoring management system. First of all, we construct a series of practical monitoring diagrams using the Cacti software and external software script following ontology theory. This enables us to completely and precisely generate an entirety network data by analysis and integration with a distributed intelligent agent mechanism (Chang & Lu, 2006). It also substantially reduces the loading of the backend server databases. Finally, we can display a high-quality quantification diagram for a dynamic network data with system webpages (Kuo, Liao, & Chen, 2005). This provides an easy, detailed solution that responds to the user's questions of network problems. It reduces the user's software expense and work-loading for the network manager, directly and easily providing a dynamic network data for all users through the system webpage. For this, we address several issues:

(1) A better Interface Agent was provided to present the user's intention or purpose descriptively.
(2) A Search Agent for the user with related domain was provided to discover and integrate a loosed network data without specific structure.
(3) An intelligent Monitoring Agent was provided to detect the network problem and display the monitoring results to the user.
(4) A high-efficiency Proxy Agent was provided as an effective substitute mechanism to shorten the system response time.

The major parts of this technique are: ontology, data integration, and proxy mechanism. We also constructed the four major parts of an efficient service network system with graphic monitoring multi-agent: Interface Agent, Proxy Agent, Monitoring Agent and Search Agent, as shown in Fig. 1. This both improves the quality of network monitoring and also provides an active network control model with intelligent network management system. The Interface Agent is the communication bridge between user and system which transferred their messages completely. It also provides the query result through the function setup of operation interface. The Proxy Agent acts as an intermediary role between the Interface Agent and the Search Agent in order to reduce the retrieval loading of the backend server databases. The Monitoring Agent immediately collected and gathered various data from different network devices, and then stored those data in the dynamic network databases with the ontology-directed format for conveniently access by the system, and outputting the monitor results directly to users. Finally, the Search Agent executed the network information gathering, considering both user-oriented and domain-related concerns with ontology-supported operation models. This is the final product of an active and intelligent network management system with multi-agent techniques.

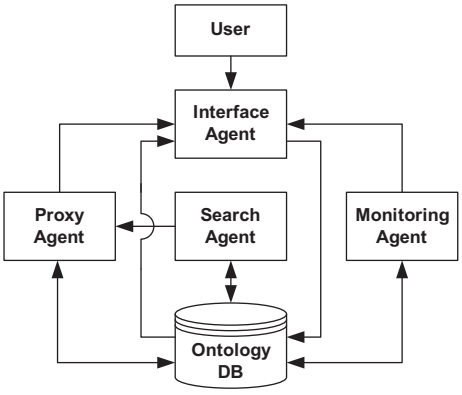


Fig. 1. Conceptual architecture of the proposed system.

The application domain of Network Performance Monitoring (NPM) monitored the base level structure of wide-area networks and local-area networks, together with network hardware equipment and their network operation status. This not only enabled the network control staff to understand the problem status in time, but also conveniently gave users sufficient information and helped share related knowledge to shortening the resolution time for network problems. The results demonstrated that the techniques implemented in this paper can both precisely recognize error alarms and also reduce recovery time (Lu, 2003) to 61% of the traditional processing time for network troubleshooting for real-time browsing, analysis, estimation, handling, and performing behavior analysis of network. The rest of the study is organized as follows. Section 2 introduces background knowledge and the development of techniques. Section 3 describes the system architecture and how it operates. Section 4 describes the system operations and their performance. Section 5 discusses related works and Section 6 offers conclusions.

2. Background knowledge and techniques

2.1. Ontology

Ontology is a philosophical theory that explores the knowledge characteristics of life and real objects. In the field of artificial intelligence it has been used to define the content of domain knowledge, express knowledge, and to solve communication problems. In the information technology field, ontology has assisted in research and development of E-commerce and knowledge management. Ontology provides complete semantic models, which includes all related entities, attributes and base knowledge in specified domains. These entities have sharing and reusing characteristics which can be used to solve the problems of common sharing and communication. Describing the structure of knowledge content through ontology can create a knowledge core for a specified domain that can automatically learn related information in regard to communication and assessment. Such a technique can even induce new knowledge. As a result, ontology is a powerful tool for constructing and maintaining information systems (Trifan, Ionescu, Ionescu, Prostean, & Prostean, 2008). Fig. 2 illustrates a part of domain ontology for the common knowledge of network protocols, which defines related basic knowledge of various information storage structures among network equipment (Lee, Yang, & Lu, 2009). This paper adopts Protégé (described later) to construct our domain ontology.

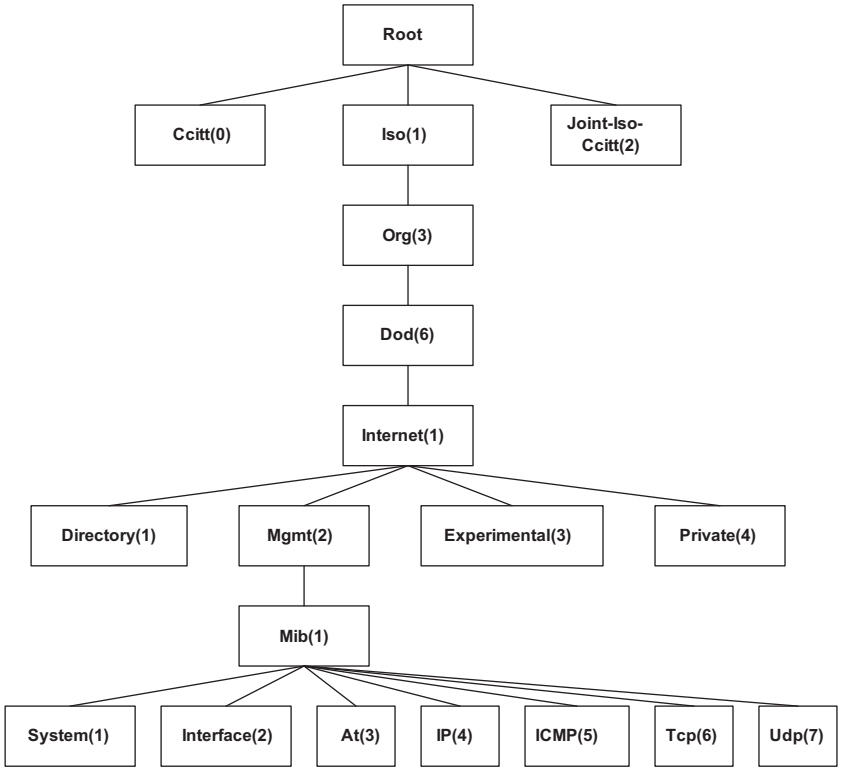


Fig. 2. Part of SNMP ontology in MIB.

2.2. Management information base

The Management Information Base (MIB) was constructed in the monitoring and monitored network equipment, which can be used to store various SNMP objects (described later). Because the different kinds of network equipment with various communication protocols have their distinctive network management models, each management model has own set of MIB objects, including an equipment's network interfaces, routing tables, IP packet transmission and receiving. In the system, we refer to this information as SNMP objects stored in the MIB Ontology DB with a canonical format so as to begin monitoring related management information of network equipment.

2.3. Protégé

Protégé (Protégé, 2009) is an ontology freeware package developed by SMI (Stanford Medical Informatics), which is one of the most important and frequently used platforms for constructing ontology. It uses multiple components such as Protégé-OWL Class, Protégé-Properties, Protégé-Forms, Protégé-Individuals, and Protégé-OWLViz to edit and make ontology and led knowledge workers to constructing knowledge management system based on ontology. Furthermore, users can transfer to different ontology formats, such as RDF(S), OWL, XML or directly transfer into a database with better support function just like MySQL and MS SQL Server. The Protégé architecture is divided into three levels, as shown in Fig. 3:

(1) The customer-design user interfaces allows users to quickly and conveniently set up specified interfaces in their own field with default interface, plug-in, and external AP.

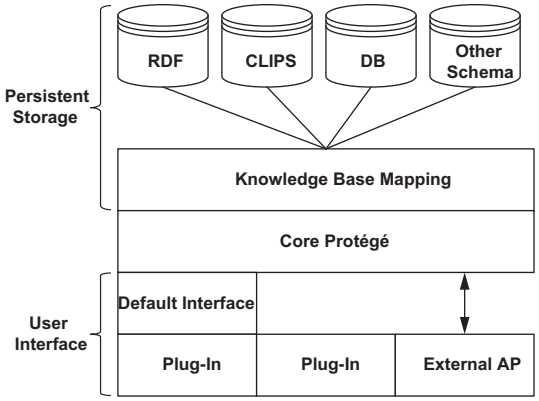


Fig. 3. Protégé architecture (data resource: <http://protege.stanford.edu/index.html>).

(2) The Core Protégé provides the main parts of knowledge management, including flexible ability for knowledge communication, powerful reuse of knowledge, powerful, and faster knowledge embedding.
(3) Persistent Storage contains the two parts of storage entities and their knowledge-base mapping, using the latter to map the domain ontologies and their corresponding instances into the storage entities.

2.4. SNMP

The SNMP (stands for Simple Network Management Protocol) is a simple and easy communication protocol for network management, which is a standard Internet protocol. It provides both a common standard for various network equipment and data for network management. This data can then be read and used for

monitoring by network management applications. In other words, there is an agent software component to be implemented for each monitored system in the network, which periodically reports the monitored information to the management system through the SNMP. The SNMP agent mainly takes care of the commands received from the network management software and reports the current operating status of the network equipment. Therefore, the agent automatically sends out a trap message to the network management software when an error has occurred in the network equipment (Yang & Yang, 2007).

2.5. Developing tools and techniques

Cacti (Cacti, 2004) is a development tool with PHP programming language to collect data by SNMP. It is integrated with various tools, such as RRDtool, SNMP, MySQL, and Apache (Liu, 2005) to create the network monitoring functions with graphic representation, which can store related monitoring information and their time-series data. It has many powerful characteristics, including the support of network protocols, with many filtering languages, and easy to review TCP communications. These features make it one of most popular professional software systems to monitor network performance. Its components can construct a network communication protocol analysis system based on ontologies, and the constructed ontology databases are suitable to be read for analysis by other network management software. It can also be imported into databases, such as MySQL, and its supported functions are more complete than other network monitoring tools. In the system presented here, we use the 0.8.7 version.

The RRDtool (Round-Robin Database tool) is the OpenSource industry standard, high performance data logging and graphing system for time series data, which can transfer those data to corresponding figures and renew those figures dynamically (RRDTool, 2009). It can be used in webpage browser to display the pictures in PNG format and those PNG pictures come from dynamic data collections that have the average utility rate of network, and peak values. RRDtool is most suitable for time series data, for which the figure's *x*-axis is time while y axis is flow rate. Detailed properties are described as following:

(1) Used as a storage format, the RRDtool can reuse data, for example, to add data into the different RRDtool files.
(2) It can draw the corresponding figures for any time division in accordance with the different requirements, for example, to construct and draw the analysis diagram of network traffic flow in time intervals ranging from one year to one half hour.
(3) Its big and powerful drafting engine can draw various quantification figures of monitoring information.

MySQL is a small-scale relational database management system with open source code that was developed by the MySQL AB Corporation in Sweden (MySQL, 2008). Currently, MySQL is popularly applied for medium- and small-scale websites to reduce the cost of website construction because of its small volume, fast speed, and lower cost. SQL (structured query language) is also a common query language for relational databases, which can be used to retrieve data from the database.

Ethereal, developed by Gerald Combs, is an open source freeware network analysis system. It is also one of the best analyzer for network communication protocol (Ethereal, 2007) since it supports both Linux and MS Window platforms and its architecture adopts the protocol tree method, as shown in Fig. 4. Since it is open source, Ethereal can quickly refresh its communication protocols and support the packet extracted file formats exported from various software package. Finally, it can go through the graphic user interface to clearly represent all network management information. In the proposed system, we use the 0.10 version.

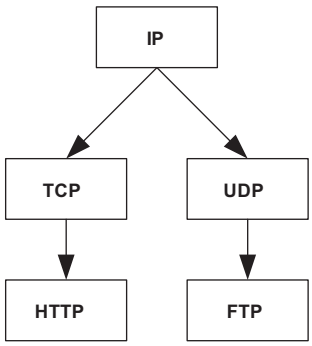


Fig. 4. Conceptual diagram of the Ethereal packet resolution protocol.

For this system, we employed the WinRcap tool built into Ethereal to define the filtering regular expression (WinPcap, 2009), including the core packet filtering, a lower-level function base of dynamic linking library, and the pattern of the high-level system function base, as shown in Fig. 5. Therefore, it can directly access the packet application interface to classify communication protocols and storage records, providing improved classification time and precision. The basic architecture of WinPcap consists of three parts: a Netgroup Packet Filter, called NPF, at the kernel level; two function bases of dynamic linking library at the user level, including a lower-level dynamic linking library and a high level and operating system independent system-independent library. At the kernel level, it is responsible for catching packets from the network interface card and then filtering and sniffering those packets. In the packet catching process, the system needs to bypass the network protocol stack to catch the transmission packets in the network. This part of the work must operate in the kernel level of the operating system and directly interact with the network driver interface, i.e., it has to define the NPF inside the WinPcap to finish the kernel level job described before. At the user level, "packet.dll" is a dynamic linking library, which provides the APIs (application programming interfaces) of the lower-level access for taking the hardware-level parameters. It also provides a public interface for the Win32 platform to solve the various problems among the different version of Windows, i.e., packet.dll can operate in the different version of Win32 platforms and it cannot be recompiled. The "wpcap.dll" is also a static function base in the catching program, which contains many system functions that are independent of the hardware models and operating system versions and which provides a high level and convenient manner to catch the packets.

The multi-agent system is a distributed environment formed by many kinds of agents. Currently, the construction trend of several software or information systems is to use many kinds of agent programs. In the multi-agent system environment, each type of agent can integrate individual techniques, knowledge, goals, and plans to cooperatively solve distributed problems. This can produce

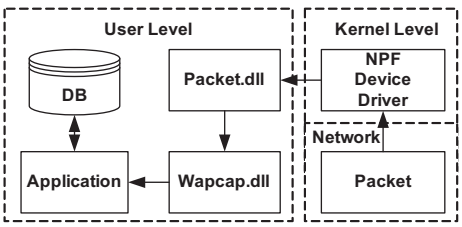


Fig. 5. Operating flowchart of WinPcap.

advanced capabilities of heterogeneity and communicative cooperation among multi-agent systems (Lin, 2005). Hence, the multi-agent systems are appropriate for finishing a specific goal and integrate several interactive agent programs to form a working group, and these agent programs even individually have the ability to solve different problems (Ren & Wu, 2008). In this study, we employ the multi-agent technique to develop the intelligent network management system with the related performance monitoring and visualized diagram drawing and refreshing. The system can surf the Internet and free-will execute following established rules and authorized scope, thus assisting the consignor to carry out the network information searching, filtering, arranging, analysis, and presentation without time or space limitations.

3. System architecture and operating processes

3.1. System development and process

Fig. 6 illustrates the system operating architecture, in which the backend system server is a Windows 2003 Server and IIS 6.0 (Yang, 2004). The intelligent agent systems and their corresponding environments were developed with Java (Tou, Lin, & Lin, 2006), which usually resides and constantly executes in the monitor hosts. We also used KQML (Knowledge Query Manipulation Language) to communicate with each collaborative agent, including communication, coordination, and division of cooperation so as to collect the dynamic packet communication protocol and corresponding data in related networks, and then store this information in backend server databases for constructing the domain ontologies. Finally, we used the freeware Ethereal and Cacti to collocate with software packages RRDtool and Net-snmp to help develop various network traffic flow and status diagrams. Those diagrams can combine the monitor figure agent with the domain ontologies to support the flow statistic analysis of domain network and corresponding communication protocols and Internet protocol analysis (Yang & Lu, 2009). The front-end client webpage can display related information and status with the PHP syntax that completed the system monitoring capabilities. This also simplified the system installation and its set-up process, substantially reducing the difficulty in practically deploying the system (Morffi, Paz, Hing, & González, 2007). Detailed working procedures of related multi-agent role, mission, and completed group are described as below.

3.2. Agent architecture and corresponding process

3.2.1. Knowledge modeling and construction of ontology database

The goal of knowledge modeling was to structure the knowledge concepts and then transform them into a form that can be processed by machines and computers. Since abstract knowledge cannot be directly transformed, it must go through a suitable transformation stage for convenient representation (Chi & Chen, 2007). Two important phases of knowledge modeling are the ontology normalization and ontology construction. The Ontology DB of this system is an ontology-based knowledge base, which mainly contains the definitions of network communication protocol, with detailed construction steps as described below.

Regular expression is a character queue to describe specified order (Lin & Chen, 2006). The descriptive style, also called pattern, can be used to search for matching patterns in another character queue and exporting those matched to the backend databases. There are two supported classes for this expression: pattern and matcher. This study uses the tool WinPcap to construct the Packet Sniffer, including the core of packet filtering, a lower-level dynamic linking function library, and a pattern of the high-level system function library to define a regular expression. The sniffer can employ regular expressions to directly access the application interface and classify those communication protocols and their storage records to improve the punctuality and precision of classification.

The ontology construction tool, Protégé, is an ontology freeware developed by SMI (Stanford Medical Informatics). Protégé not only is Java-based in support of version of various platforms, but also can plug-in the visualized hierarchical diagram (TGVizTab) to extend its construction function for conveniently viewing the consistency of the ontology contents. Its most advantageous feature is that of uses multiple components to edit and develop an ontology, allowing knowledge workers to construct a knowledge management system based on ontology.

Firstly, the system employs the OWL (object windows library) Web ontology language to describe the information definition of the MIB. Then, it goes through the support of the constructed MIB Ontology DB to take the structured classification, using the

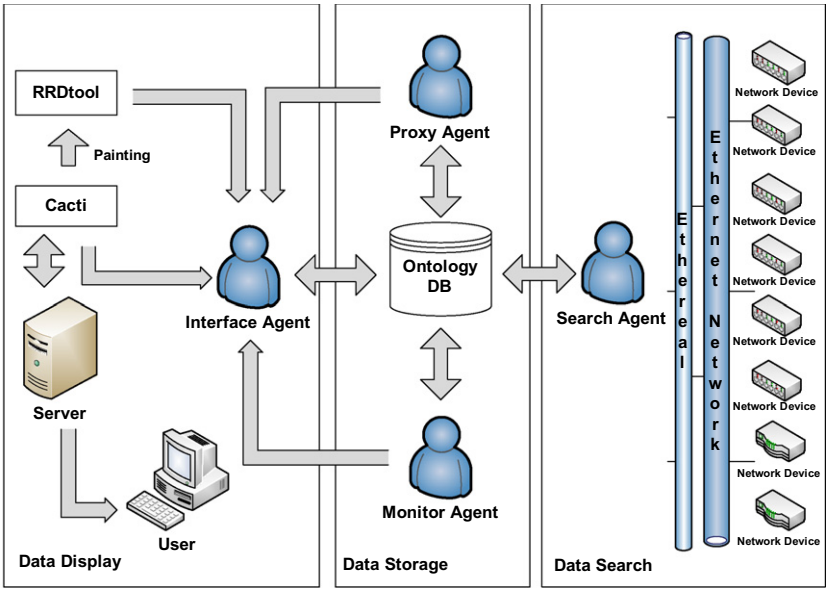


Fig. 6. System operating architecture.

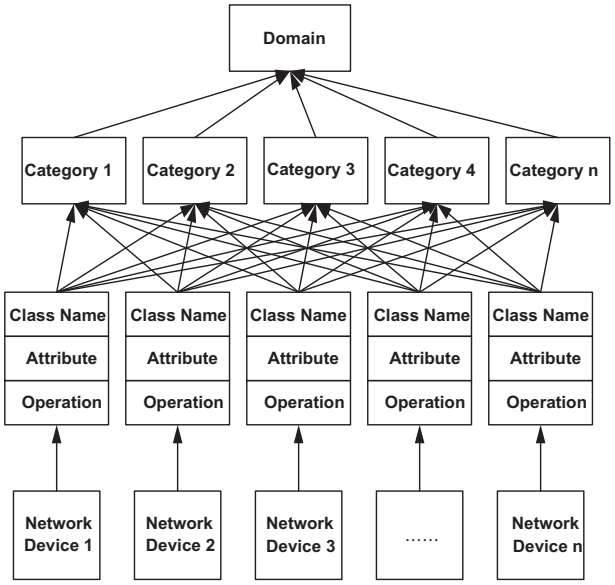


Fig. 7. Conceptual diagram of the MIB Ontology DB.

hierarchical concepts of the MIB to present the relationships among each hierarchy. These constructed ontology structures can be used to analyze the meaning contained in various packets and be stored in the corresponding ontology databases to form the differently classified groups. Its conceptual diagram is shown in Fig. 7.

This figure explains the conceptual ontological diagram formed by the collected information in a network through the support of the Packet Sniffer in the Search Agent. Domain means an application that can execute the SNMP network component, which can gather the corresponding network packets and traffic flow information. Category[1...*n*] means the sub-classes that can execute the defining information and managing functions in the specific MIB. The Class Name means the sub-class of the Category that contains related contents of every packet in the corresponding protocol. For example, an MIB contains some sub-classes, including the main agent information together with its allocated parameters, response to the administrator requirement, and corresponding warning alarms. The above defined information also has some sub-classes, including System, Interfaces, Address Translation, IP, ICMP, TCP, UDP, and EGP, and the individual title number of every sub-class is shown in Table 1.

This study gathered several common communication protocols and their application services currently used through the support of the MIB. Their characteristics and specifications of message format in both the corresponding packet header and its content of the application layer were used to generalize the matching characteristic words for using in the packet classification. Based on this, we designed the MIB Ontology DB. That database has the OSI seven-layer contents, dynamic status records, and information maintained on communication protocol. It can generalize the matching characteristic words for use in surveying the packet headers and OSI seven-layer contents according to the characteristics and specifications of several common communication protocols and their application services in order to provide enough matching categories. The system used the Ethereal tool to provide a Statusfully Content-Based Classification Engine as the base of packet classification to satisfy the requirements of convenience and elasticity. Having enough matching classes to describe various communication protocols and application services, and integrating those category statistics to establish the system database enabled the system to search for exact information. In other words, the system employed the constructed MIB Ontology DB to support the Search Agent in carrying out the collection of related network protocol packets and classification processing. Inside the Search Agent, the ontology guided the domain packet collection. For classification, the ontology goes a step further to subdivide the classes according to their characteristics and it designs many class patterns to be the bases of protocol classification. Therefore, we only construct a domain ontology to support the Search Agent, but have to set up several types of ontologies to support the classification function. Through the support of the MIB Ontology DB, the system can automatically determine the connection with monitoring information. The ontology database constructed in this study first gathers statistics and analyzes related concepts of network protocols and then constructs a corresponding ontology database, as described below.

Table 1
Corresponding descriptions between the class and its title number.

Class	Title number	Contained message
System	(1)	Operating system of the Host or the router
Interfaces	(2)	Each network interface and its measured amount of the network traffic flow
Address translation	(3)	Address translation
IP	(4)	Statistics on IP grouping
ICMP	(5)	Statistics on the received ICMP messages
UDP	(6)	Algorithm, parameters, and its statistics
EGP	(7)	Statistics on the agreed amount of network traffic flow of the external gateway

Firstly, we collect the characteristic words from related concepts of network protocols and their corresponding packet formats, as shown in Fig. 8. The system is based on those characteristic words and classifies their corresponding standard packet format according to the classification patterns to enable convenient monitoring information citation by the Search Agent. In other words, if the matching process fits in with the monitoring concept of the corresponding communication protocol and its packet format is exact, the system determines the classification target of the gathered information. It then stores that information under that classification target in the MIB Ontology DB for the collection of real information to enhance the precision of the system's determination of information.

The system employed the Protégé OWL editor (Chen & Yang, 2006) to construct the MIB Ontology DB to finish the second stage of ontology construction, is shown in Fig. 9. Finally, support of the XML file (conveniently fixing the semantic errors if necessary) was used to transform that constructed file into the backend server database for conveniently access by the related system components. In addition, the system was arranged in pairs of the ontology and match rules, including the network protocol, source and destination IPs, as well as source and destination communication ports, to support all of the agent operations.

3.2.2. Interface Agent

In the overall system, the Interface Agent acts as administer and adjuster for handling all user queries, including those from the frontend user requirement services to the backend adjustment of individual agents. To obtain the related equipment information it is necessary to go through this agent's coordination to simultaneously carry out each agent's work, that is to synchronously monitor all actions of database information access. Hence, it can propose the hypothesis to look after both sides of the data variation and system resource for providing the related information services. Fig. 10 illustrates the detailed architecture of the Interface Agent and the relationship diagram among other agents. Inside

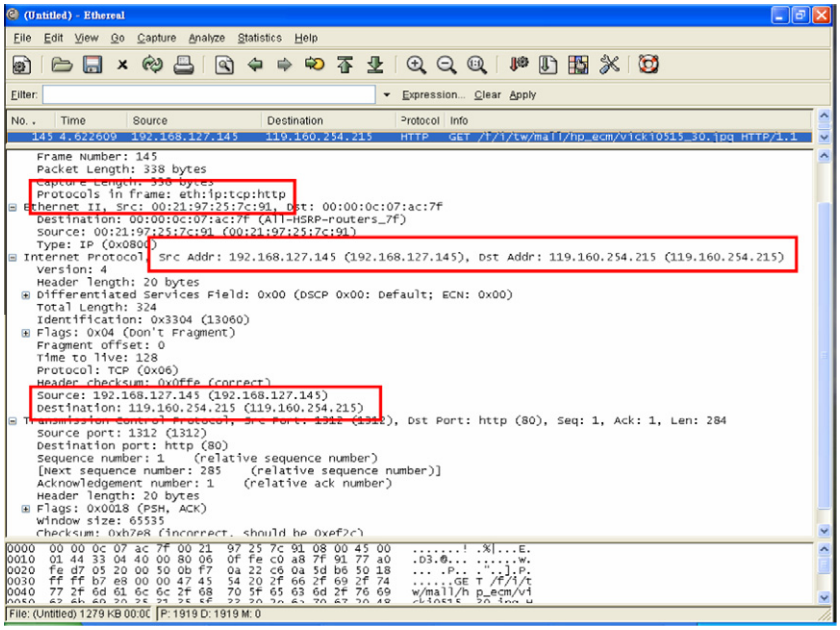


Fig. 8. Network protocol and corresponding packet content.

the agent, its modules contain the User Login for confirming the user's login process, the User Manager Program for managing the user's function access authority, the Personalized Web for presenting the personalized webpage, the Painting Component Model for drawing related quantification figures of dynamic information, and the Webpage Processor for predictably loading and processing the related webpages of visualized figures.

When the Interface Agent gets a user login request, it turns to the backend User Profile Database to search for the user-related information supported by the User Manager Program. This can certify the user's identification and authority for carrying out the classifying management on user accounts, including the following four kinds of records:

(1) *Auth*: To certificate the user account identification.
(2) *Account*: To be the account management of decertification, for example, to check whether the deadline for user account and password are overdue.
(3) *Password*: To certify the user password identification.
(4) *Session*: To associate the user with those works needed to execute the access services from beginning to end, for example, taking down the information on the directory mounting and limiting the resource usage to the user.

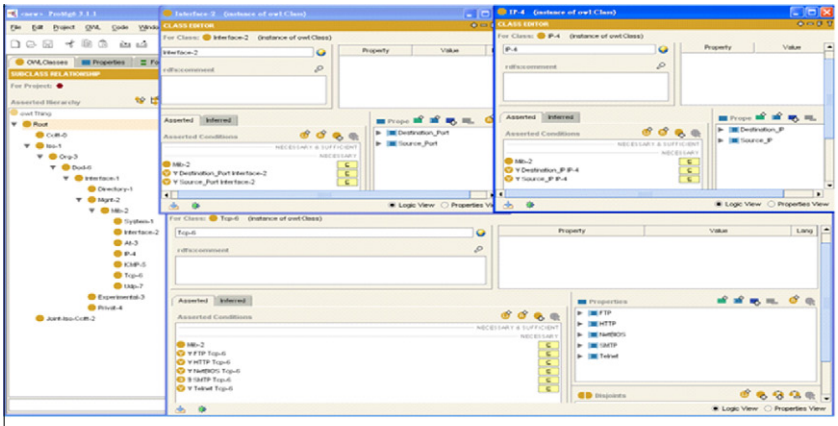
The User Manager Program can search for the proper browsing authority to fit the user login in accordance with the record of the above database. It validates execution testifying and account certification for the application program according to different user authorities. Once the success certification is archived, the User Manager Program returns the certification result to the Personalized Web for calling the personal webpage mechanism to produce the corresponding personal webpage (Yang, 2007). In other words, the system can be based on the login authority to filter out the suitable resources for the user to browse the up-to-date and qualified resources. Fig. 11 illustrates its operating process.

The Painting Component Model employs the RRDtool module package of Cacti to periodically gather the related data of traffic flow of network equipment and draw the corresponding figures through the support of the Monitoring Agent and the Proxy Agent, as shown in Fig. 12.

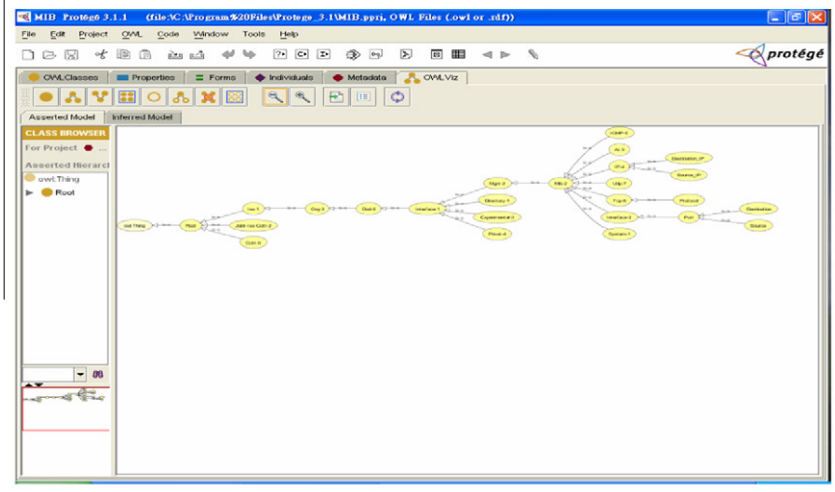
The statistical information produced daily is stored in the domain Ontology Database (Chinese Open Systems Association, 2003). The system carries out messages receiving from statistical information returned by the Monitoring Agent and the Proxy Agent, such as IP traffic flow ranking, date and time related traffic flow information. After the Relation Data has accomplished the connection with diagrammatic network traffic flow, the system proceeds to handle Database Updating with the Cacti figure template of traffic flow according to the received information and corresponding PHP programs to draw out the information integration of the analysis on communication protocols, and IP traffic flow ranking. Fig. 13 illustrates the part of the diagram display in PHP. Then the Painting Component cyclically draws out the related monitoring diagrams. Finally, the system presents the integrating monitor webpages of the network system through the processing of the Interface Agent and shows out related real-time visualization diagrams of the network with the support of the Webpage Processor, including the diagrams of network traffic flow and equipment status of the network. Each piece of network equipment in those diagrams can be periodically refreshed in accordance with their real-time states, and there by improving the beneficial results of information integration on network monitoring and related qualities of information services.

3.2.3. Proxy Agent

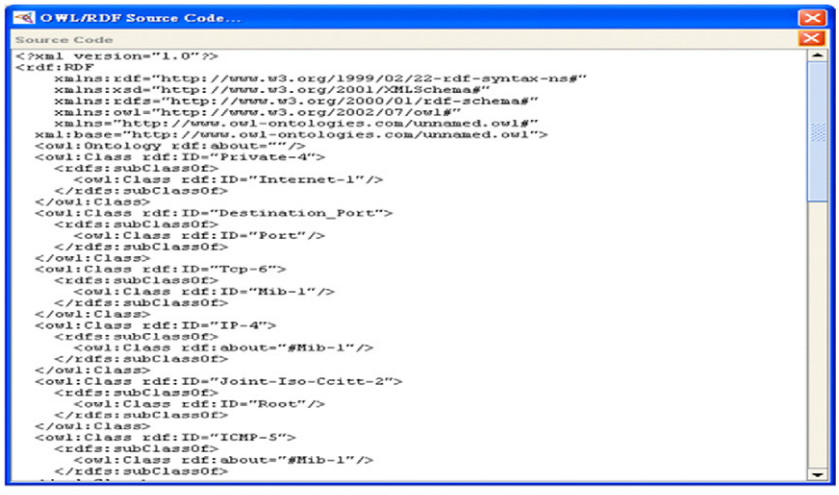
The traditional proxy function always caches the retrieval information against the previously query into the system registers. When the system wants to get the same information for the next query, it can retrieve that information from the system registers directly. But this approach could increase the waiting time of the first user for browsing. This proxy architecture is intended to improve the functions of the traditional architecture and develop an active mechanism for the Proxy Agent. It consists of the four modules, as shown in Fig. 14, including the Proxy Space, which is responsible for autonomously storing the prior retrieval information; the Refresher, which is responsible for automatically refreshing the system information according to the definition of time interval in the Data Display Model; the Data Display Model, which is responsible for precisely showing the system information; and the



(a) Connecting step of ontology construction



(b) Ontology hierarchical diagram



```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Private-4">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Internet-1"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Destination_Port">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Port"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Tcp-6">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Mib-1"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="IP-4">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Mib-1"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Joint-Iso-Ccitt-2">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Root"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ICMP-5">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Mib-1"/>
    </rdfs:subClassOf>
```

(c) Part of Ontology XML

Fig. 9. Domain ontology construction with Protégé.

Scheduler, which is responsible for periodically retrieving that information from the system databases.

The proxy function combined with the Proxy Agent mechanism through the support of the Data Display Model, which consists of the Data Connection, the Data Integration, the Data Updating, and the Data Record, is shown in Fig. 15. The Scheduler periodically calls the Data Connection in the Data Display Model and autonomously sends an information query to the Ontology DB with the data query syntax of the standard SQL. This can parse out the information resource locations in that query and accordingly carry out the information reading from those database resources. The execution steps are as follows:

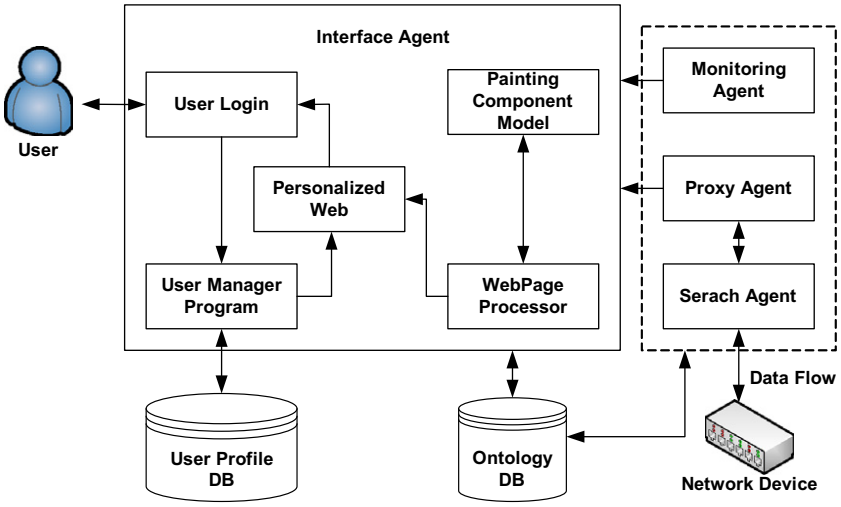


Fig. 10. Architecture of the Interface Agent.

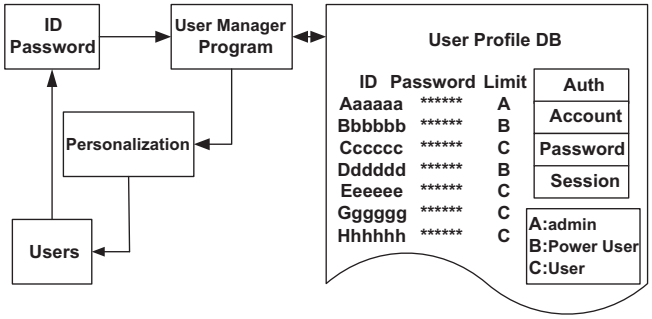


Fig. 11. Architecture of the user account certification.

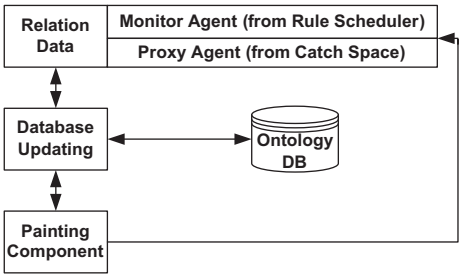


Fig. 12. Flowchart of the Painting Component Model.

(1) *SQL Command Execution: To* execute related SQL commands.
(2) *Database Object Access: To* search for related data tables.
(3) *Data Access: To* execute the database reading.
(4) *Data Extracting: To* obtain the necessary information.

The queried information, such as IP (including Source and Destination), communication protocols, network traffic flow, date and time are individually delivered to the Data Integration in the Data Display Model to handle the integration of associated information and prepare the information presentation, for example, the corresponding Destination IP to its Source IP and the communication protocol conclusion. Then that information was passed into the Data Record as a new information record through the support of the Data Updating, and that completed information integration was returned back to the system database to be the historical query base for maintaining the database robustness. Finally, the

```
<?
    $corder=@$HTTP_GET_VARS['color'];
    $width=@$HTTP_GET_VARS['width'];
    $height=@$HTTP_GET_VARS['height'];

    if ($corder=="") {$corder='000000';}
    if ($width=="") {$width=20;}
    if ($height=="") {$height=20;}

    if ($width<=0) {$width=1;}
    if ($height<=0) {$height=1;}

    $im=imagecreate($width,$height);
    $back_color=$corder;

$black=imagecolorallocate($im,hexdec(substr($back_color,0,2)),
hexdec(substr($back_color,2,2)),hexdec(substr($back_color,4,2)));

    imagefill ($im,0,0,$black);
    imagepng($im);
    imagedestory($im);
?>
```

Fig. 13. Part of the diagram display in PHP.

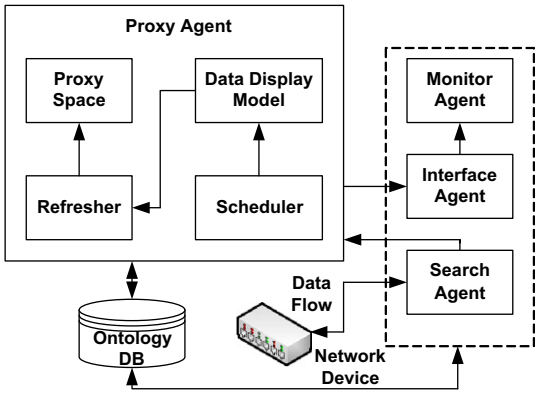


Fig. 14. Architecture of the Proxy Agent.

system automatically refreshed the internal information via the component Refresher and temporarily registered at the Proxy Space to wait for an information request come from the Painting Component Model in the Interface Agent. The series of processes in the monitor information integration could increase both the

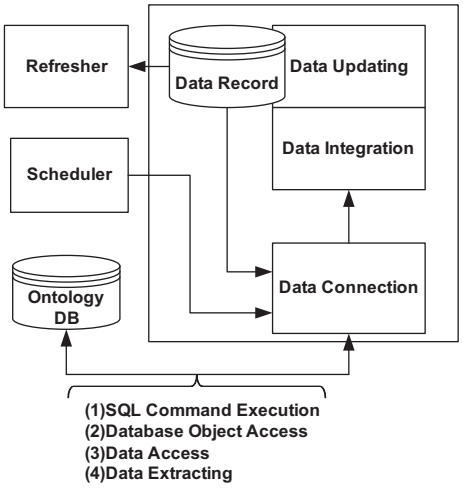


Fig. 15. Architecture of the Data Display Model.

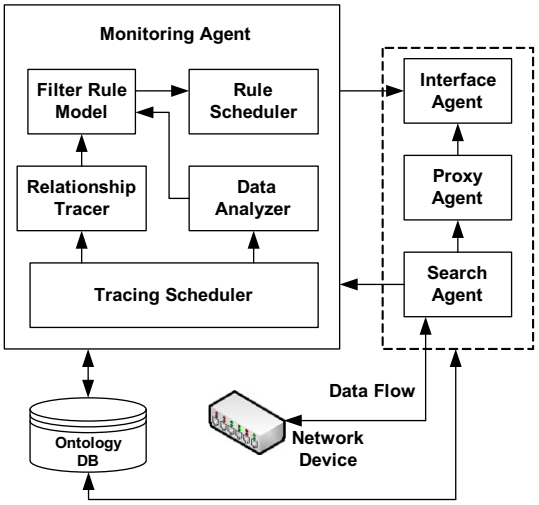


Fig. 16. Architecture of the Monitoring Agent.

speed of the monitor information browsing and the efficiency of related processing for clearly reducing the access time of the monitor information. Therefore, the Proxy Agent can practically shorten the query time from the user to the system database in accordance with the true information requirement of the user.

3.2.4. Monitoring Agent

The traditional associative checking procedure for producing warning messages is a one-by-one to check to deal with equipment in the network through an application program. This procedure not only has a complicated and time-consuming calculation process but also to waste its processing time by reduplicating completed parts. Consequently, this paper proposes a new generation of warning service using the related monitoring and calculating mechanisms of the Monitoring Agent. This is automatically integrated the useful information in advance and uses centrally processing to autonomously provide up-to-date and effective monitoring information according to the user requirements. The Monitoring Agent can simultaneously monitor the network equipment and their traffic flow to effectively and efficiently handle the dispatching and coordination process. This considers hypotheses of various information variations and limitations of system resources to actively and effectively monitor information services (Lin & Wu, 2008). This system monitors the traffic flow and network equipment as shown in Fig. 16. The Monitoring Agent contains: the Tracing Scheduler, which is responsible for periodically taking the database reading; the Relationship Tracer, which is responsible for associating the network communication ports; the Data Analyzer, which is responsible for analyzing and retrieving the traffic flow information; the Filter Rule Model, which is responsible for doing the warning calculation; and the Rule Scheduler, which is responsible for periodically sending those monitoring results to the Interface Agent. First, the agent uses the Tracing Scheduler to periodically read the necessary information from the system databases. Because the information records of the system databases are very diverse, the agent must be supported by the Relationship Tracer and Data Analyzer to analyze and abstract the necessary information chunks from the different system databases, such as the interface communication ports of the network equipments, the corresponding traffic flow in the network, and the time that information occurs in order to integrate this information. The Monitoring Agent also is supported by the Filter Rule Model, as shown in Fig. 17, for the various kinds of Data Reading using the Relationship Tracer and Data Analyzer. This includes the Data Analysis and

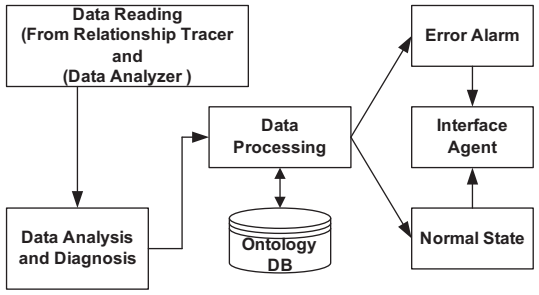


Fig. 17. Architecture of the Filter Rule Model.

Diagnosis, Data Processing, Error Alarm, and Normal State, detailed operating process as shown below.

First, the Filter Rule Model obtained the state message of each communication port in the network from the support of the Relationship Tracer, such as whether the function situation is Up or Down. It then acquired related information from the support of the Data Analyzer, for instance, the individual traffic flow of a communication port in the network equipment. The agent analyzed and diagnosed these two aspects of information to determine the final monitoring results based on the user requirements with the system weighted mechanism. For example, the normal green state means the average response time was less, equal or less than 40 ms; the red warning alarm means the average response time was greater, equal or greater than 81 ms or there was no response at all; and between these ranges, the yellow warning alarm appeared. Finally, the agent presented the proper warning message through the support of the Interface Agent according to the final warning. This approach not only both quickly produced the processing information but also stored it in the system databases at the same time. This has two advantages: one is to develop historical records to autonomously make analyze traffic flow to conveniently use as the follow-up information for queries, and the other is to automatically produce a suitable response in accordance with the processing message designed by the network administrator in advance.

The following is an example on the operating illustration of the Monitoring Agent. The experimental environment is the Host and related network equipments in the information center of a Branch of the Bureau of National Health Insurance in Taiwan, including two L2 switches and one L3 switch. The L2 switch was connected

with the user terminal; the L3 switch was connected with the Host and system databases; and in addition the backbones of the L2 and L3 switches were connected with each other and their communication ports had their own IP locations. Finally, the communication ports of the L3 switch connected to the L2 switch were established as the corresponding mirror ports so the system could periodically extract the communication protocols and related IPs. The upper-left corner in Fig. 18 illustrates a demonstration of the Monitoring Agent at 11:28 a.m., May 11th, 2009. In this figure, the system displays a yellow warning alarm on an L2 switch with the average response time of 80 ms. Clicking the yellow icon can go a step further to show the advanced monitoring diagram with a yellow state in a communication port, as shown in Fig. 19. The corresponding equipment name of the yellow icon can be clicked to find its information in the trend diagram of the network traffic flow appositive rising, as shown in Fig. 20. The user can proceed to select and display the one-minute traffic flow diagram at the time point of this event, as shown in Fig. 21. The agent found the abnormal traffic flow at the above time point and immediately started protocol analysis on network traffic flow to confirm what kind of protocol resulted in the abnormal network event. It went through that analysis to find that 74% of the traffic flow to the Host came from the FTP protocol, detailed on the right-bottom corner in Fig. 18, and the system confirmed the reason for the traffic flow jamming. Finally, the network administrator can inquire about the IPs that caused the large volume of network traffic by clicking the left-bottom corner in Fig. 18 to show the billboard diagram of IP traffic flow for searching for the corresponding source IPs and their destination

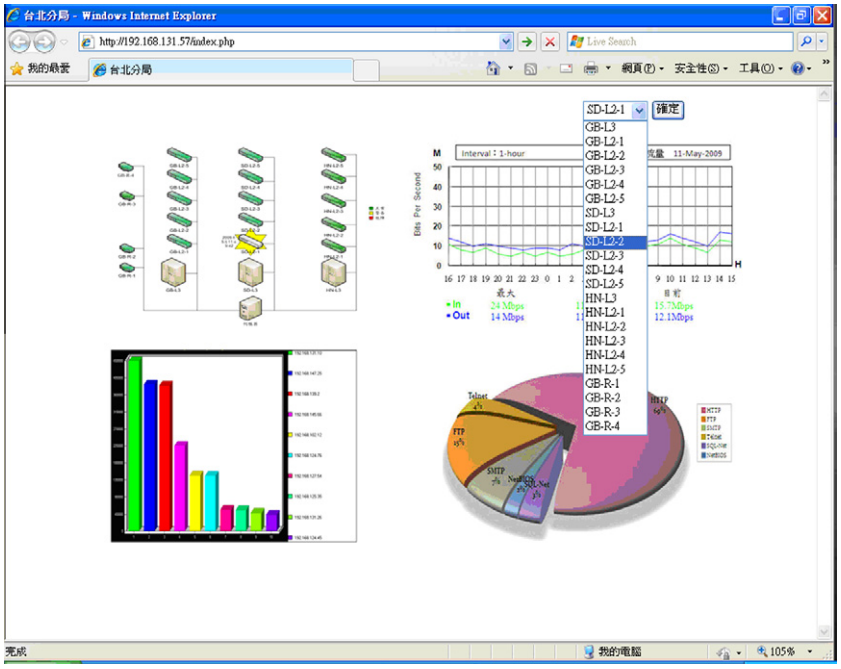


Fig. 18. Example on the warning message of the L2 switch in the network.

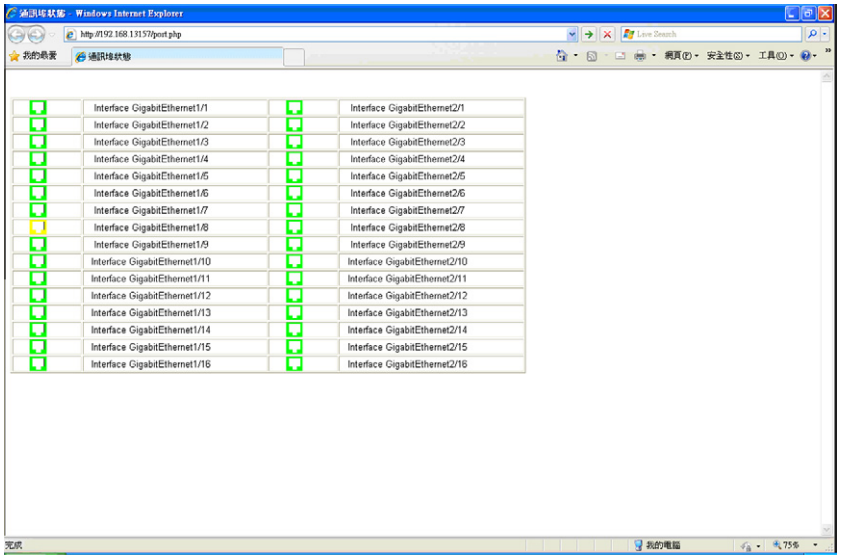


Fig. 19. Communication port warning alarm in the advanced monitoring diagram.

Fig. 20. Trend diagram of the network traffic flow.

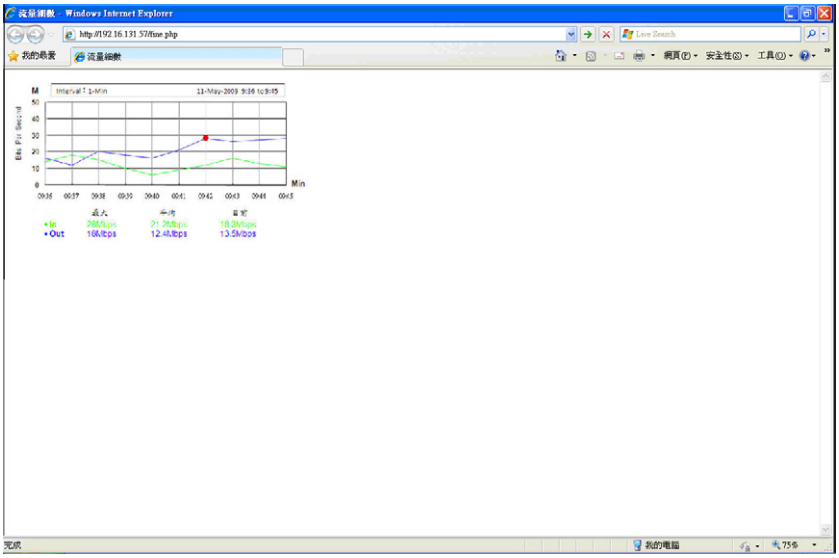


Fig. 21. One-minute traffic flow diagram.

IPs, as shown in Fig. 22. Thus the network administrator not only can easily determine the kind of network malfunction, such as a large volume of traffic and a broken network connection, but also can quickly determine the abnormal event points to quickly separate them to deal with concrete matters to shorten the recovery time of the network malfunction (Chen, 2004). This approach reduces the monitoring information quantity of users and makes the network monitoring system more efficient.

3.2.5. Search Agent

To gather the necessary information on network monitoring is to take actions that suit local circumstances and produce the information format and content for supporting the Monitoring Agent are completely various too. The Search Agent was established in the SNMP core, which employed related domain ontologies and related function libraries of Cacti and Ethereal to observe the connection status of the overall network and its equipment. Fig. 23 illustrates its complete architecture, including the Linking Status, the Packet Sniffer, the Data Gathering, and the Protocol Analyzer. The operations were divided into the packet collection and the traffic flow collection. The former starts using the function of communication port monitoring in the network equipment. It employs the Packet Sniffer that was produced by the packet gathering functions built-in the Ethereal to collect all transmission packets in the network. Finally, it analyzes the communication protocol and its corresponding IP with the support of the Protocol Analyzer. The latter collects the traffic flow information of the whole network with the support of the SNMP Get communication protocol of the Cacti (Shin, Jung, Cheon, & Choi, 2007). The functions and operations of the related components are described as below.

The Packet Sniffer can receive the transmission packets come from the network through the support of the WinPcap. The Ethereal-based Protocol Analyzer can analyze the protocol, which classifies the transmission packets into what kind of protocol according to those packets belong to what kind of data flow. It can classify the packet data flow based on the five fields of the

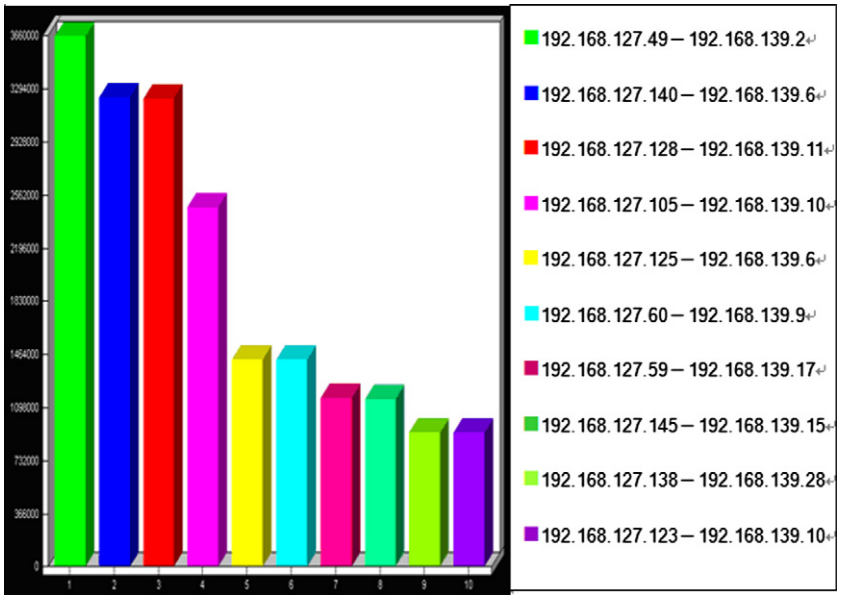


Fig. 22. Billboard diagram of IP traffic flow.

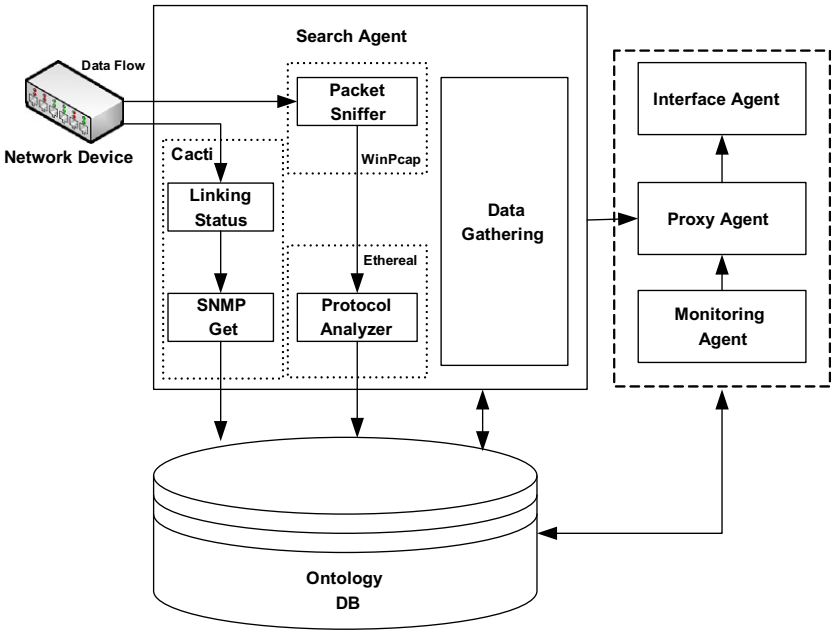


Fig. 23. Architecture of the Search Agent.

packet header, including Source IP, Destination IP, Protocol, Source Port, and Destination Port. The agent based on the headers and contents of those packets to carry out the filtering action for discriminating using Protocol, IP, and Port, and can then store the basic information of each data flow and related information statuses of network equipment into the Ontology DB. In addition, the Linking Status periodically uses the Windows Ping command to check the latest status of the network equipment to maintain the connection checking on the communication ports. At the same time, the agent uses the SNMP Get communication protocol of Cacti to periodically collect the network traffic flow information, which records the connection information and gathering the traffic flow information. That network monitoring information can be stored into the Ontology DB with the support of the SNMP Get.

The agent can strengthen the functions of information analysis and corresponding database connection through the support of the Data Gathering based on the identification code of each data flow. It then goes to the Ontology DB to search for the records of corresponding data flow (Yang & Tseng, 2007), such as the corresponding IPs of the SMTP, HTTP, TCP, UDP, and ICMP. It can directly retrieve the defined monitoring information classified according to the Ontology DB, based on the integration of related monitoring information to the corresponding IP, and it also simultaneously refreshes the database information. The agent can survey the packet header to find out the fitting rules in accordance with the display information defined by the Proxy Agent. Finally, it can deliver the complete monitoring information to the Proxy Agent to conveniently output the related monitoring information to users through

the Interface Agent. In real time the user can precisely observe related network status, including network equipments, use of communication protocol and operating information and related statuses of IP and network bandwidth.

The following is an illustration to explain how the Search Agent works. It first introduced characteristic words to determine the kind of protocol because each protocol must register own dedicated words. The sub-node belonging to the characteristic word can agree to provide the distinguishing identification each other, such as the TCP.Port = 21 means the protocol is an FTP. To carry out the protocol classification, detailed in Table 2, this study defines the five fields of characteristic word in the Ethereal description language based on the packet header, including Source IP, Destination IP, Protocol, Source Port, and Destination Port. For example, a value of the Protocol field is FTP and its corresponding value of sub-node equals TCP.Port = 21. Therefore, when a TCP packet with the Port number 21 arrives, the Protocol Analyzer searches for its corresponding characteristic word with the support of the MIB Ontology DB until the exact sub-node with TCP.Port = 21 was found. Detailed steps are shown in Fig. 24. In addition, the agent went through the SNMP Get command in the Cacti to gather the In and Out traffic flow of each communication port in network equipment to collect the network traffic flow information and accordingly store that information in the system databases.

Table 2
Match parameters of Ethereal description language.

Match	Description
Source	Source IP
Destination	Destination IP
Protocol	Protocol
Sport	Source Port
Dport	Destination Port

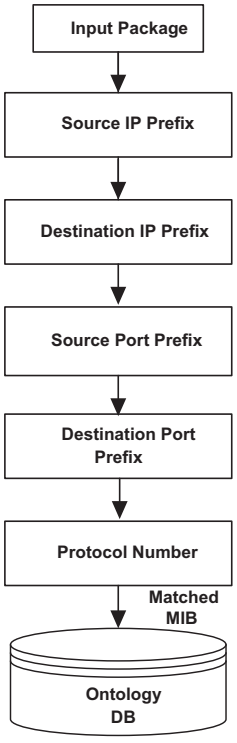


Fig. 24. Packet decomposing flowchart.

The interaction processes and their goal summaries of this multi-agent system are described as follows:

(1) When the Interface Agent receives a user query, the Proxy Agent plays the role of mediator between the Search Agent and the system databases, which deals with the information temporarily retained in advance. This can provide the proxy mechanism to enhance the efficiency of the query cache services and thereby clearly reduce the query response time.
(2) The Monitoring Agent is responsible for assisting users in monitoring network information, which can execute the corresponding information services and retrieve related information. Its service description format can directly acquire the query information from the system databases. Once finds information matching the user requirements, it immediately transmits that information to the Interface Agent to dynamically present the network statuses.
(3) The Search Agent obtains related network information from the monitoring equipment through the real network connections and that information will become the data resources to construct the system databases. The network status information is divided into two parts: the real-time monitoring information and the continuous historical information. They are classified and separately stored in the backend databases. The system employs the backend databases to collect and store related network information in advance and then provides the information resources to the Proxy Agent for related query cache services.
(4) The active and intelligent network management system is a multi-agent system directed by an Interface Agent. The Interface Agent not only plays the role of communication bridge between the user and the multi-agent system, but also distributes the user query requirement to the system and accordingly assists the user in presenting the outcome of analysis on monitoring information. The user can both use the system for common queries on problems between the equipment and traffic flow, and also go through the system to make hierarchical queries, to present the analysis results at a higher level and/or the specific details for replying to the user (Huang, Chiu, & Tsai, 2007).

3.3. System functions and limitations

The system used the Ethereal-based Protocol Analyzer to periodically monitor and gather network packets. In addition, the Search Agent combined with the Cacti monitor software to periodically use the SNMP objects to obtain the monitoring information on the network equipments and store that information in the corresponding backend databases. The Interface Agent can integrate and present that monitoring screen (Lai & Guo, 2006). To simplify the system operations and provide a user-friendly interface, the system adopted the client webpage representation so users could interact directly with the system through the browser in the local end. Then the system read the related information from the backend databases and accordingly presented the visualized information to users to achieve the goal of a real-time network monitoring interface that was fast, precise, and more user-friendly. The function design of related systems is described below.

The system used the Cacti monitoring software combined with related components of the Monitoring Agent to construct a service architecture with dynamic network monitoring manner to integrating network connection monitoring, network malfunction checking, traffic flow analysis, and monitor platform maintenance. This quickly provides various types of exact information for

autonomously monitoring and also reduces the response time to deal with the network malfunctions, thus improving the efficiency and level of network operations. When the monitoring network resources had problems, the system can autonomously stand on the built-in SOPs (Standard Operation Procedures) by contact with one or more related events, thereby responding and presenting those events to users in time through the support of related agents. In summary, the system uses the Search Agent to periodically evaluate the monitoring network equipment and accordingly obtained many kinds of information related to the operating situations of the network equipment and stores this information in the backend databases. Then the Monitoring Agent actively detects and triggers related specific events or status notices based on the above SOPs. Finally, the system uses the Interface Agent to quickly present graphic warning alarms for the network (Chiang, 2005; Marano, Matta, Willett, & Tong, 2006). The advantages of this approach are determined by: there is shorter detection time after a network event occurs, the processing time and effectiveness are improved with little network loading; it can precisely and quickly achieve active network monitoring. In addition, certification for the authority and identification of each system function used different queries to authorities according to their corresponding identifications for authorized control of the users' authority to browse the information (Tokihiro, Masayuki, & Takuji, 2006).

4. System display and verification

To obtain clear understanding of all network situations, the homepage of the system was designed in a hierarchical manner consisting of four elements: the status diagrams of equipment, the trend diagram of the traffic flow, the billboard diagram of IP traffic flow, and the traffic flow analysis diagram of protocol in the network. The second layer of this hierarchy is the detailed situation of the network equipment, including all related information of the network equipment, connection status, and communication protocols; as shown in Fig. 18, which is the main screen of the system. The detailed operations of related sub-systems are described below, respectively.

4.1. System log-in

The user used only the client-end browser to connect with the log-in webpage of the system and the system immediately entered the certification screen to request the user account and corresponding password, as shown in Fig. 25, for browsing the related system webpages after the successful certification. The user cannot install any additional software to use the system.

4.2. User interface

The system used a visualized monitoring mode to conveniently control the authority to browse webpages, dividing users into User, Power User, and Admin. A User can only browse the first-layer monitoring screen provided by the system after the login; the Power User can enter the second-layer monitoring screen to browse the advanced diagram presented by the system; the Admin can browse both of above two screens and also can set up the monitoring network equipments and related users authorities. Those set-up functions are detailed as below.

(1) *Device*: To add or delete the IPs of monitoring network equipments, as shown in Fig. 26. After the IP adding or deleting operations, the system can display the corresponding IP in the IP Address List under this webpage. If the user wants to inquire about how many communication ports are monitored by the system, he/she can select the corresponding query button in the Interface field, as shown in Fig. 27.
(2) *Admin*: To add or delete the user account and corresponding password, as shown in Fig. 28. After the user adding or deleting operations, the system can display the corresponding account, password, and authority to browse system webpages in the List under this webpage.

4.3. Billboard of network traffic flow

Fig. 22 shows the billboard of network traffic flow that can tell the user which sever occupies the most bandwidth in the network. It also can show the cause of this large amount of bandwidth on the network among the corresponding servers when the user clicks the corresponding webpage icons for easy display of the advanced bandwidth status, as shown in Fig. 29.

4.4. Traffic flow of network protocol

Fig. 30 illustrates the usage situation of the network bandwidth for each protocol application and the system also analyzed which application was over-loading the network. This provides an easily understood diagram of the connecting mode and the utility rate of its bandwidth for convenient observation and querying by users so that users can understand which software system occupies the most bandwidth in the network.

4.5. Tendency of network traffic flow

Fig. 31 provides a graphic diagram of whole network traffic flow. The system can analyze any time period of network traffic flow for current or previous day and use the warning message for the maximum bandwidth threshold to actively monitor the network traffic flow. When there is a network interruption or a fully-loaded bandwidth, an early warning is automatically issued by the system. Hence, the user could handle this situation as soon as possible so as to avoid the network malfunction and track down the entire occurrence of the event over time.

4.6. Status diagram of network equipments

The system can display the status diagram of monitored network equipment in the fundamental structure as shown in Fig. 32. The real-time responding statuses are shown by corresponding connected icons when the malfunctions occurred in those devices or their connections. The system shows those statuses with a color corresponding to their malfunctions: red[1] color for a malfunction, yellow for a warning, and green for a normal situation. In addition to showing exceptional events in the network equipment with corresponding colors, their occurrence times are also marked on the diagram. This allows users to go a step further and click the corresponding icons to understanding the detailed states of the network equipment, as shown in Fig. 33. It also provides the traffic flow from level 2 to level 4 to each port in the entire network. This tells the advanced user which part of the fundamental structure was a network problem and also shows which business has been effected (Li et al., 2008).

4.7. Warning monitor

The exceptional malfunction states of many monitoring items can be determined by comparing the threshold value between the

[1] For interpretation of color in Figs. 18, 19, 32 and 33, the reader is referred to the web version of this article.

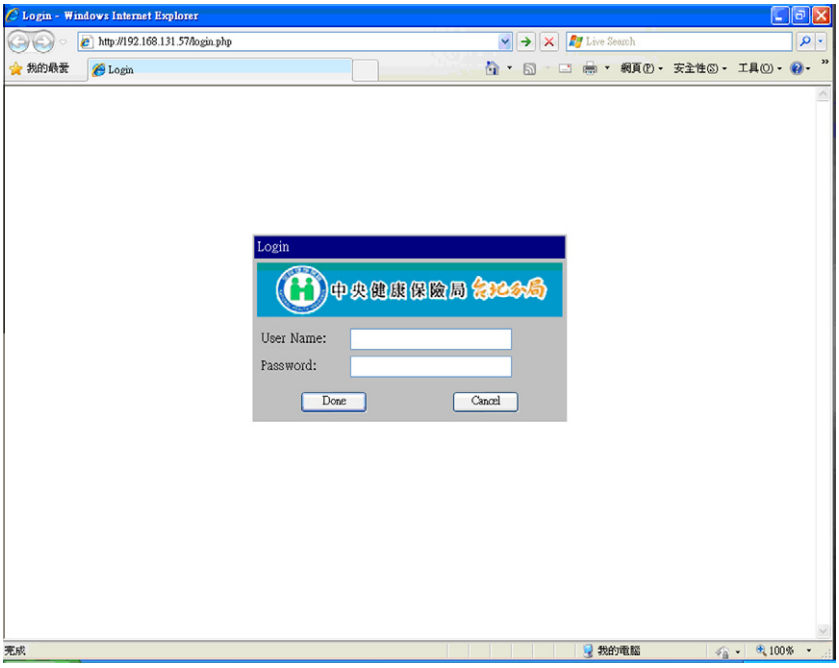


Fig. 25. System log-in screen.

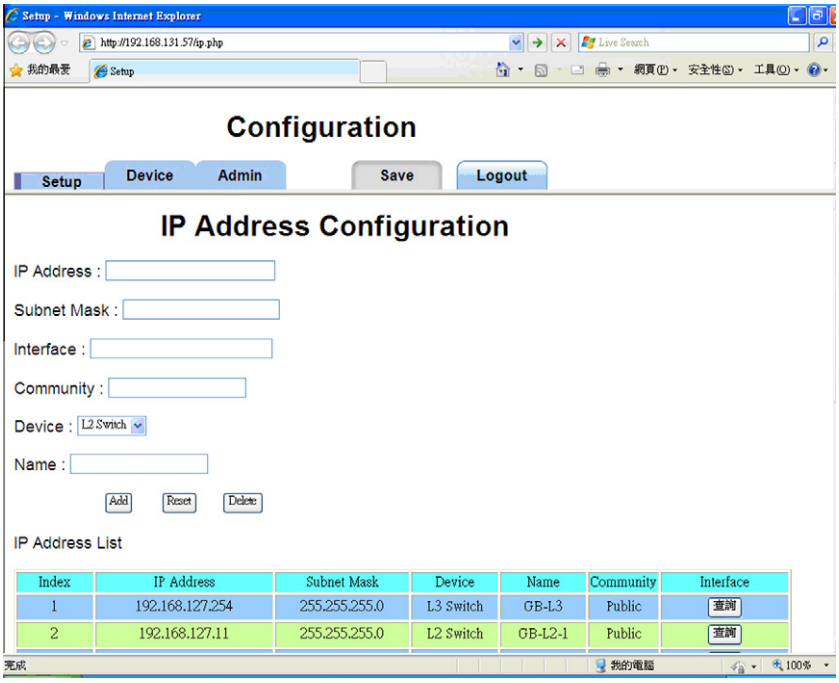


Fig. 26. Screen for adding or deleting monitoring devices.

measured data and the standard value. For example, the network traffic flow in the abnormal and performance management of a network monitor system. However, the measured data has a range of reasonable variance due to the differing environment, timing, and user behavior. In the Window operating system, the system can diagnose the network connection state and its quality with the 'Ping' instruction. This instruction checks the network connection state with the Echo function of ICMP (Internet Control Message Protocol) according to the RFC 792 (Request For Comments) regulation unquestionably. In other words, a small packet was sent to an IP address in the network, and a waiting query was occurred for the responded packet and data continuously. Actually, the sender receives a complete response packet and corresponding data if the network connection is good condition and the network equipment is working properly. The time for those packets route of travel is calculated below. In this system, we propose the filtering condition to analyze the network traffic flow with a standard value consisted of the high elastic analysis on traffic flow, modularized monitoring parameters, and composite monitoring conditions (Baker et al., 2008). The formula is shown as follows:

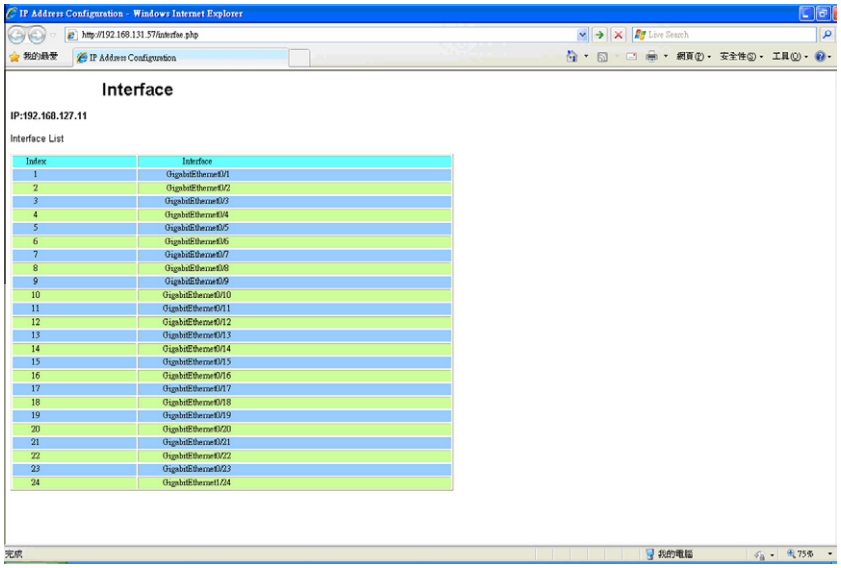


Fig. 27. Screen with the selected query button in the Interface field.

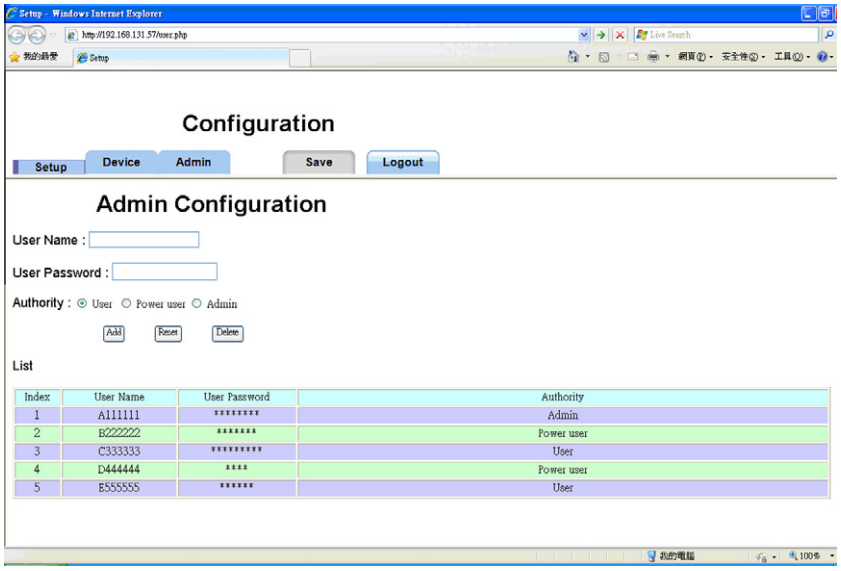


Fig. 28. Screen for the Admin.

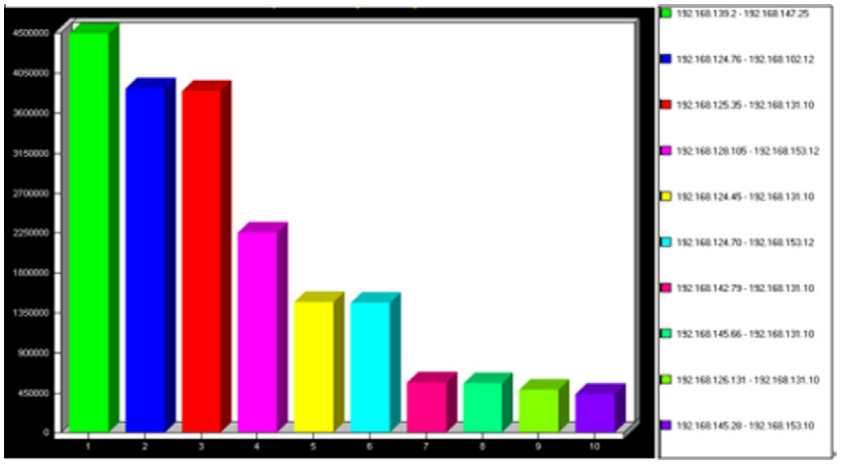


Fig. 29. Advanced analysis of network traffic flow.

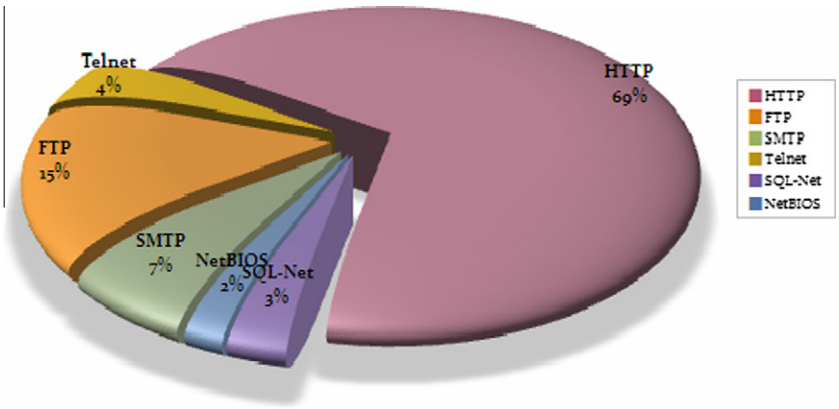


Fig. 30. Traffic flow analysis diagram of network protocol.

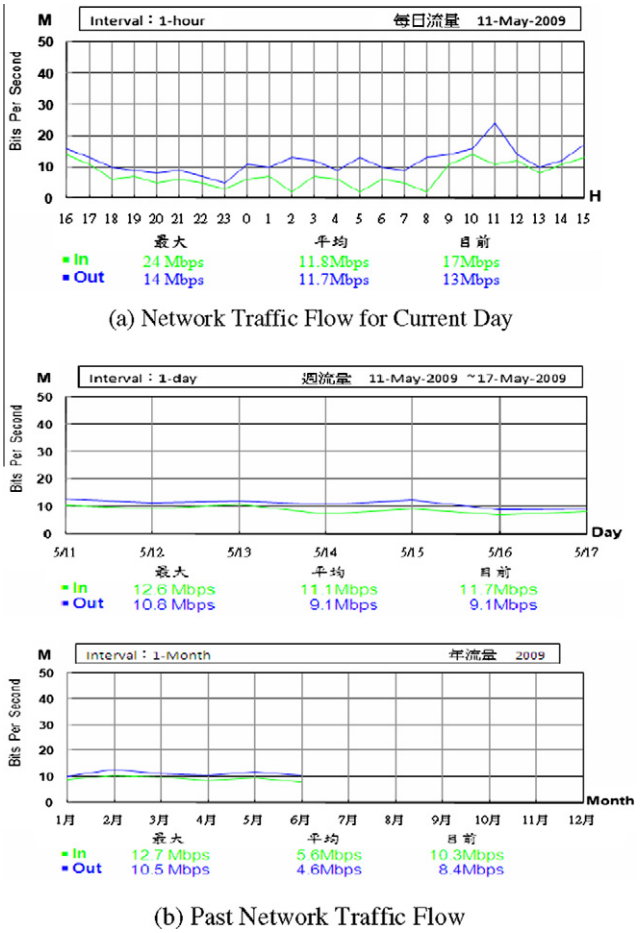


Fig. 31. Tendency diagram of network traffic flow.

[Reply time (1) + Reply time (2) + Reply time (3) + Reply time (4)/4] = Average
Reply time: the response time from the network equipment
Average: the average time after calculation

The normal green state occurs while the average response time is less than or equal to 40 ms; the red warning alarm occurs when the average response time is greater than or equal to 81 ms or there was no response at all; and otherwise, the yellow warning alarm appears. This can be used by the system to determine whether or not the existing network performance is good, and the system can conveniently proceed to eliminate the network malfunction in accord with the average response time.

This study uses many techniques to develop an active, intelligent, and multi-agent network management system to construct a centralized catching system based on the network traffic flow. The system uses the necessary information on the network monitoring process to provide a graphic quantification figures to show related network information in a fast and convenient user interface webpage (Loreto & Eriksson, 2008). The related intelligent agents usually reside and constantly execute in the monitor nodes, which can immediately determine an abnormal network behavior when the network malfunctions, and accordingly provides the warning alarm to show the node with exceptional network behavior (Tseng & Liu, 2007). This allows users to easily evaluate the current network state to precisely understand the statistics on network traffic flow. It also reduces the monitoring cost for bandwidth usage of a business network so as to provide the necessary accurate information on network monitoring (Moraes et al., 2008a).

4.8. System performance verification – a practical application

4.8.1. Correctness of warning alarms determining

In the experiment, we evaluate the traffic flow warning alarm and packet collection and classifying work. The network topology of the experiment environment was a closed local area network. We gathered all packets of the Host at the information center of a Branch of the Bureau of National Health Insurance in Taiwan between May 8, 2009 and May 21, 2009. The system operated in coordination with the corresponding user information base built up in advance to analyze and understand the usage situations of related IP within the network region for providing the two-way, interactive, and real-time monitoring query system with the Web manner, displaying all monitored and analyzed results, as shown in Fig. 34.

In this figure, the system displays the distribution diagram of the traffic flow warning alarms at each time point. The *X*-axis shows the date and time, while the *Y*-axis shows the average response time in milliseconds for a round-trip between the system and the monitored Host. Compared with the advanced monitoring diagram of network traffic flow, as shown in Fig. 35, the system can precisely show that all of above time points indeed had higher traffic flow that affected transmission efficiency of the monitored network. The precision rate of warning alarm reached almost 100%.

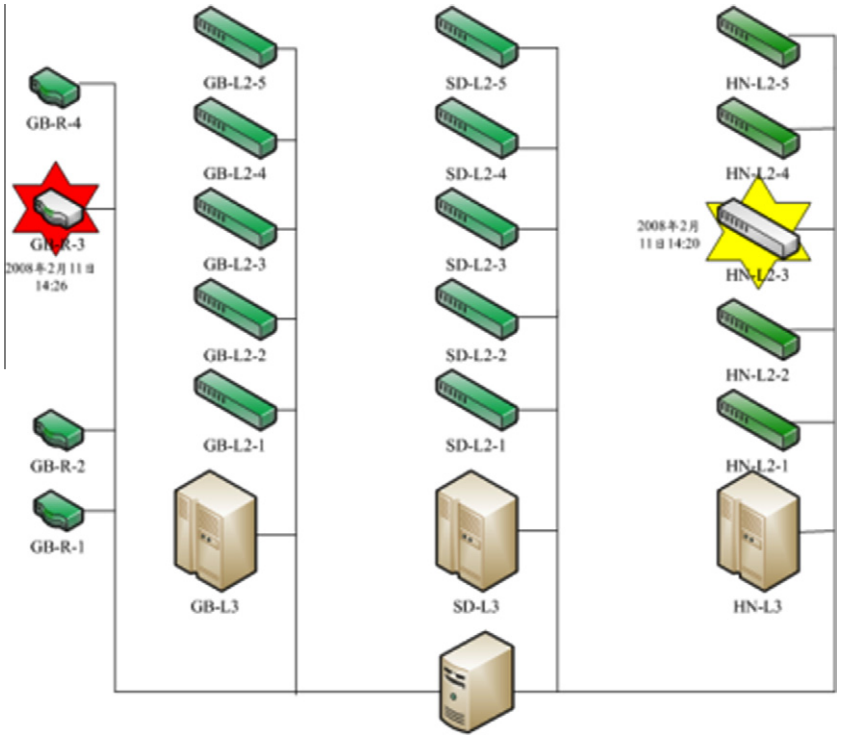


Fig. 32. Status diagram of network equipment.

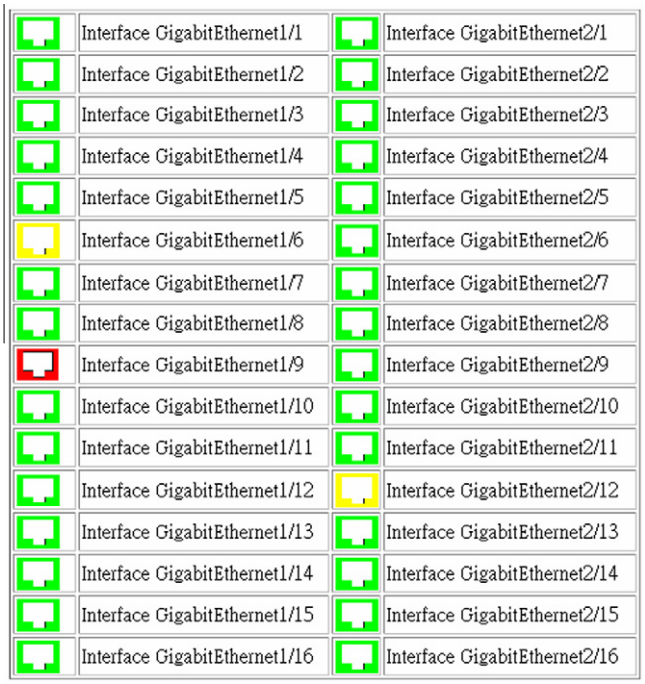


Fig. 33. Advanced monitoring diagram of network equipments.

4.8.2. Time and efficiency on the system processing

In its early stages, the network malfunction processing was made through an oral report of the problems by the user. However, administrators often encountered unclear statements by the users had difficulty determining the precise kind of network problems. Thus they could only depend on the network equipment topology and the SOP to sequentially check step by step, which took a long time. To handle a large-scale network topology, the time needed to fix problems did not encourage the administrators to resolve them. Thus the traditional processing time and efficiency cannot already satisfy current practical requirements. Since the network staff spends so much time to check network problems, they cannot analyze and deal with related network problems in time. In the two week period required to introduce this system into the practice and testing environment, we trained users to use the system, including how to view the network equipment status and the traffic flow information so users could understand their own network environment. If network malfunction occurred, the user can use this system to understand what kind of problem caused this event. The network staffs can go through the system to determine what problem occurred in which network node, and then eliminate related network situations. Fig. 36 combines three items of common problems in the two week introduction period, including the network equipment malfunction, the network connection problem, and the problem of large traffic flow. To analyze processing time, the processing time of network equipment malfunction was reduced to 67% of the usual (30/45 = 67%); the processing time of the network connection problem was reduced to 50% of the usual (10/20 = 50%); the processing time of the problem of large traffic flow was reduced to 67% of the usual (20/30 = 67%); and finally, the average processing time was reduced to 61% of the usual. This system was deployed and operated easily and it precisely and effectively provided warning alarms when a network malfunction appeared.

4.9. Related systems comparison

There are several characteristics of this study, such as the overall fundamental structure of monitoring the network equipments, the visualized diagrammatic icons for monitoring the operating statuses of network equipments, the dynamic monitoring of network status, and the real-time understanding of the malfunction information (Valliyammai & Selvi, 2008). A comparison table for the network management software in marketing to explain our system's pros and cons is shown in Table 3. The WhatsUP system

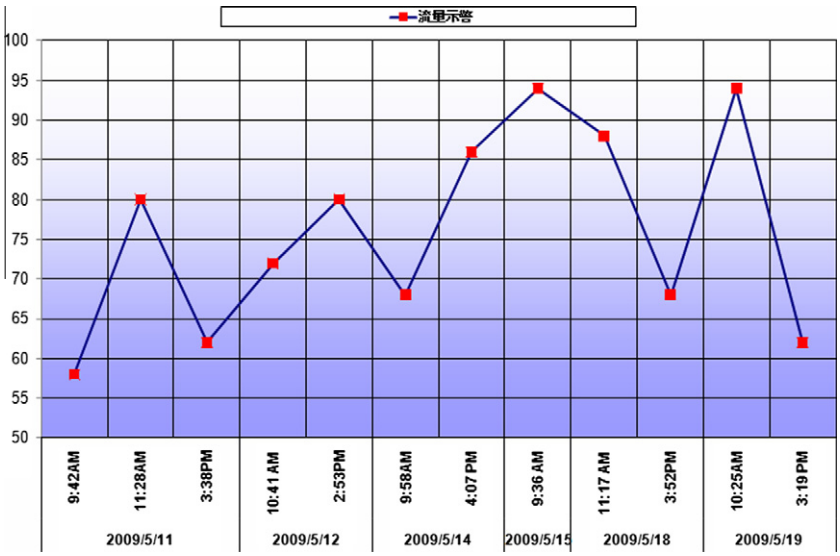


Fig. 34. Distribution diagram of traffic flow warning alarm.

can provide neither remote IP monitoring nor visualization diagram interface. The NetVCR system worked only on the SNMP protocol. The last column of Table 3 shows the approach proposed in this paper, which can provide the complete, effective, and precise network management functions.

5. Related works comparison

Computer network as a driven-force drags its customers to share more and more resources. On the other hand, however, managing such network recourses is a challenging job for an IT (Information Technology) expert as well as difficult in view of human. The agent is an autonomous entity which senses and acts upon an environment and directs its activity towards achieving goals (or programmed goals) by learning or using knowledge. It is a trend and important technique in computer science to integrate network technology with single or multiple agent systems. Here are some examples as such systems. Manzoor & Nefti (2009) proposed an agent-based system to monitor resources over a network, commonly known as campus area network and employed a multi-agent system to ensure proper system operation by watching for inconsistencies in user activities, node level activity, internet monitoring, and system configuration. Wu et al. (2009) presented a multi-agent design method and system evaluation for wireless sensor network based structural health monitoring (SHM) to validate the efficiency of the multi-agent technology. This used the cooperation of six different agents for SHM applications in which the distributed wireless sensor network can automatically allocate SHM tasks, self-organize the sensor network, and aggregate different sensor information. Tzou, Lee, and Jeng (2009) proposed a methodology to exploit ontology-based biological knowledge for network analysis and an agent-based framework to support distributed computing and speed up the analysis. Moraes et al. (2008b) proposed a new approach to monitor the performance of advanced Internet applications based on the use of an expert system, which inferred from a domain ontology called MonONTO. This ontology collected the main concepts and their relationships in the following sub-domains: quality of service of advanced applications, network performance measurements, and user profiles. After analyzing the network monitoring technology and IA (Intelligent Agent) technology, Wu, Zhao, and Ye (2008) proposed a method based on IA for complete dynamic network monitoring. This was developed by a cross-platform language Python, including two monitoring modes: automatic and manual, and a monitored wait mode to realize the performance of dynamic network monitoring more reliably. Yoshida, Shomura, and Watanabe (2007) proposed a visualization technique of network status that uses a frequent itemset mining algorithm to find important phenomena in the network and showed that a simple interface with the proposed technique can visualize both the ordinal network status and various security incidents. All of these systems were developed with specific or purchased programming languages/platforms; however useful systems can also developed using free software, can't they?

This paper proposes a system that obtains information through the cooperation and coordinate of an intelligent multi-agent software. In addition, the system also provides warnings after analysis to monitor and predict some possible error events among controlled objects in the network, and such operating mode. This yields an active and intelligent network management system with ontology-supported multi-agent techniques based on free software. Li et al. (2008) presented a virtual-machine oriented architecture for network traffic monitoring and analysis, and under the architecture it is logically divided into one host, one virtual machine monitor, and multiple virtual machines. Its cross-platform feature, supported by the virtual machine capability, however, provides another level of flexibility in system development. Jiang (2008) also proposed a low-end embedded devices direct access Jini network solutions to aim at characteristic network with specific embedded systems. The two views of mentioned above deserve more attention.

6. Conclusions and discussion

This study uses many techniques to develop an active, intelligent, and multi-agent network management system using free software to construct a centralized caching system based on the network traffic flow. The system obtained the information through the cooperation and coordination of the intelligent multi-agent software. In addition, the system also provided warnings after analysis to monitor and predict some possible error events among controlled objects in the network. This provides an active and intelligent network management system with ontology-supported multi-agent techniques based on free software. This technique derived from an ontology combined with related free software, Ethereal and Cacti, stored the operating information of network management in the backend database. It also integrated with the

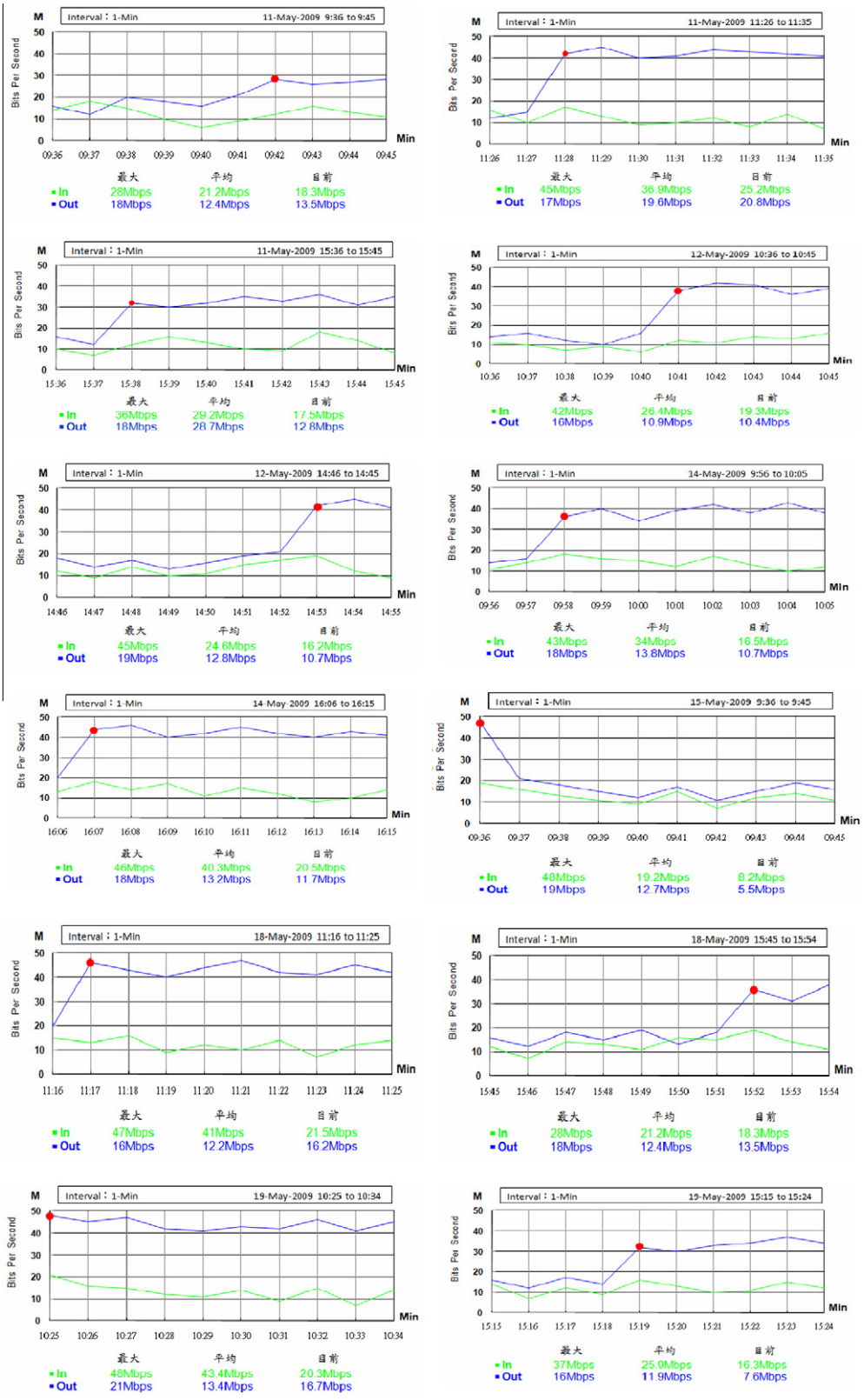


Fig. 35. Advanced monitoring diagram of network traffic flow.

intelligent agent technique to effectively enhance and improve the network monitoring performance, and accordingly present related quantification figures of dynamic information. The preliminary experimental outcomes of the system prototype showed that the techniques implemented in this paper could not only precisely recognize error alarms but also reduce the recovery time to 61% of the traditional processing time for network troubleshooting for real-time browsing, analyzing, estimating, handling, and engaging to behavior analysis of a network in use. This can be of great help for both network users and administers.

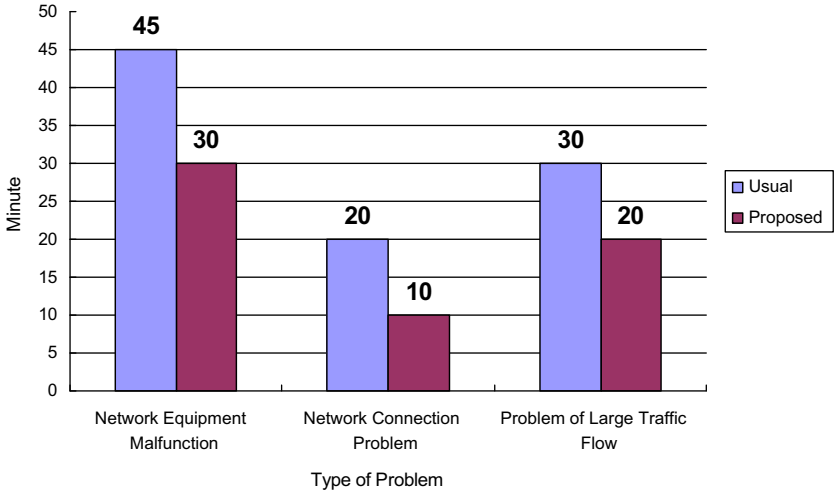


Fig. 36. Comparisons between traditional method and proposed system.

Table 3
Function comparison for network management software.

Features or functions	WhatsUP	NetVCR	Cacti + template
SNMP monitoring	V	V	V
Single operation with multi-query	V	X	V
SNMP monitor for multi-platform	V (S/W)	X (H/W)	V (S/W)
Real-time information display	V	X	V
Remote IP monitoring	X	X	V
User defined SNMP warning values	V	V	V
Visualization diagram	X	X	V

Most general network management systems on the market need to have the system detection value set up by the network manager at the beginning. The system we propose can not only automatically determine the cause of malfunctions but also completely analyze the malfunction reasons for the network manager. Furthermore, it can present related quantification figures of dynamic information of a graphic network monitoring system, which tries to develop a more intelligent system with integration, intellectualization, and distribution.

(1) *Integration*: The system can be combined with a business information system to integrate and handle various brands and different domains of network equipment into a single system. It can provide a single interface with powerful and easily operated functions to instantaneously monitor the whole network. The system can go through a Web-based interface to provide the real-time and unlimited space-time properties of the network, periodically monitor the network equipment, immediately find network problems, and use the least resources to accomplish effective network management.
(2) *Intellectualization*: The system introduces intelligent agent techniques into the network management to periodically collect the network status, storing and gathering statistics on network management information, executing and replying to commands from the network manager to check abnormal network situations. It can not only instantaneously observe the network but also carry out its tendency analysis, thereby responding to the related problems of the network system.
(3) *Distribution*: The system not only relies on a single network management system but also has the double advantage of a centralized and hierarchical structure. It can go through a single integration system of network management to apply the access and operation of all network information, network warning, event reports, and network management. The network monitoring is distributed to each local management client, and the remote monitoring function can be reached through the remote management supported by the backend server.

The adaptability of the proposed system is enhanced by its use of the open source code. Users can precisely realize and control abnormal or irregular phenomena to determine problems in either network equipment or network circuits using the easy graphic interface based on the periodic monitoring analysis by this system. This alleviates the work load for network staff, and also reduces the cost of related network maintenance and professional training. In the future, we plan to expand our system to an improved and easy interface that considers both operating and management levels. This can allow the network administrator to assign and handle all network equipment through the visualization interfaces to then display and print out analysis reports and monitoring figures.

Acknowledgements

This partial work was supported by the National Science Council, Taiwan, ROC, under Grants NSC-99-2221-E-129-012 and NSC-99-2623-E-129-002-ET, and the Ministry of Education, Taiwan, ROC, under grant Skill of Taiwan (1) Word No. 0990045921s.

References

Baker, D., Nodine, M., Chadha, R., Chiang, C. J., Gottleb, Y., & Hsu, C. P., 2008. Computing diagnostic explanations of network faults from monitoring data. In *Proceedings of IEEE military communication conference, CA, USA* (pp. 1–7).
Cacti (2004). *The complete RRDTool-based graphing solution*. Available at <http://www.cacti.net/> (visited on Nov. 5th, 2009).
Chang, Y. H., & Lu, T. Y. (2006). A study on an adaptive learning architecture with intelligent agents. In *Proceedings of 2006 Taiwan network conference, Hualien, Taiwan*.
Chen, C. H. (2004). *An agent-based network abnormal monitoring mechanism*. Master thesis, College of Computer Science, Asia University, Taichung, Taiwan.
Chen, C. N., & Yang, C. H. (2006). Developing a system of cybercrime investigation and prevention based on OWNS. In *Proceedings of 5th conference on information technology and applications in outlying Island, Kinmen, Taiwan*.
Chi, Y. L., & Chen, Y. C. (2007). Developing an information retrieval system with user-recognition based ontology. In *Proceedings of 18th international conference on information management, Taipei, Taiwan*.
Chiang, T. W. (2005). *SNMP management*. O'Reilly Media Corporation Taiwan Branch, Taipei, Taiwan.

Chinese Open Systems Association (2003). *The standard operation procedure on setting up database servers. COSA-SOP-2003-006, version 1.00*. Available at <http://www.oss.org.tw/doc/92doc1/SOP-MySQL.pdf>, (visited on Nov. 5th, 2009).

Ethereal (2007). *The world's most popular network protocol analyzer*. Available at <http://www.ethereal.com/> (visited on Nov. 5th, 2009).

Huang, C. W. (2008). *Introduction to communication systems of network management*. Taipei, Taiwan: Chinese Taipei Components Certification Board.

Huang, C. J., Chiu, M. S., & Tsai, Y. L. (2007). Design and developing of monitoring platform with integration network services. In *Proceedings of 2007 Taiwan network conference, Taipei, Taiwan*.

Jiang, X. (2008). Low-end embedded devices access Jini network design. In *Proceedings of 2008 international conference on advanced computer theory and engineering, Phuket, Thailand* (pp. 1057–1061).

Kuo, S. Y., Liao, F. P., & Chen, K. L. (2005). *Network management: Concepts and practice, a hands-on approach*. Taipei, Taiwan: GoTop Book Corporation.

Lai, S. C., & Guo, W. C. (2006). Measurement of network quality based on application layer. In *Proceedings of 2006 Taiwan network conference, Hualien, Taiwan*.

Lee, D. L., Yang, S. Y., & Lu, S. H. (2009). Developing an ontology-supported information recommending system for scholars. In *Proceedings of 2009 conference on information technology and applications in Outlying Island, Kinmen, Taiwan*.

Li, Q., Hao, Q. F., Xiao, L. M., & Li, Z. J. (2008). VM-based architecture for network monitoring and analysis. In *Proceedings of the 9th international conference for young computer scientists, Hunan, China* (pp. 1395–1400).

Lin, C. D., & Chen, Y. F. (2006). *Database management and applications*. Taipei, Taiwan: Chuan Hwa Book Corporation.

Lin, Y. C. (2005). *A study and implementation of mobile agent with SNMP in distributed system*. Master thesis, Dept. of Information Management, Chinese Culture University, Taipei, Taiwan.

Lin, W. C., & Wu, L. C. (2008). Design on user-based behavior mining and abnormal detection mechanism. In *Proceedings of 7th conference on information technology and applications in outlying Island, Penghu, Taiwan*.

Liu, P. F. (2005). *FreeBSD integration applications on heterogeneous systems and network management*. Taipei, Taiwan: Unalis Book Corporation.

Loreto, S., & Eriksson, G. A. (2008). Presence network agent: A simple way to improve the presence service. *IEEE Communications Magazine, 46*(8), 75–79.

Lu, Y. C. (2003). *Dynamical E-learning system based on intelligent agents*. Master thesis, Dept. of Information Management, National Pingtung University of Science and Technology, Pingtung, Taiwan.

Lu, C. F. (2005). *Network planning and management*. Taipei, Taiwan: XBook Corporation.

Manzoor, U., & Nefti, S. (2009). An agent based system for activity monitoring on network – ABSAMN. *Expert Systems with Applications, 36*(8), 10987–10994.

Marano, S., Matta, V., Willett, P., & Tong, L. (2006). Cross-layer design of sequential detectors in sensor networks. *IEEE Transactions on Signal Processing, 54*(11), 4105–4117.

Moraes, P. S., Sampaio, L. N., Monteiro, J. A. S., & Portnoi, M. (2008). MonONTO: A domain Ontology for network monitoring and recommendation for advanced internet applications users. In *Proceedings of IEEE/IFIP network operations and management symposium, Salvador da Bahia, Brazil* (pp. 116–123).

Moraes, P. S., Sampaio, L. N., Monteiro, J. A. S., & Portnoi, M. (2008). MonONTO: A domain ontology for network monitoring and recommendation for advanced internet applications users. In *Proceedings of 2008 IEEE network operations and management symposium workshops, Salvador Da Bahia, Brazil* (pp. 116–123).

Morffi, A. R., Paz, D. R., Hing, M. M., & González, L. M. (2007). A reinforcement learning solution for allocating replicated fragments in a distributed database. *ComputaciÓn y Sistemas, 11*(2), 117–128.

MySQL (2008). *The world's most popular open source database*. Available at <http://www.mysql.com/> (visited on Nov. 5th, 2009).

Protégé (2009). *Stanford University Protégé teaching website*. Available at <http://protege.stanford.edu/> (visited on Nov. 5th, 2009).

Ren, W., & Wu, X. (2008). Intelligent detection of network agent behavior based on support vector machine. In *Proceedings of international conference on advanced computer theory and engineering, Cairo, Egypt* (pp. 378–382).

RRDTool (2009). *RRDTool logging & graphing*. Available at <http://oss.oetiker.ch/rrdtool/> (visited on Nov. 5th, 2009).

Saturday (2008). *Cloud computing*. Available at <http://mmdays.com/2008/02/14/cloud-computing/> (visited on Nov. 5th, 2009).

Shin, K. S., Jung, J. H., Cheon, J. Y., & Choi, S. B. (2007). Real-time network monitoring scheme based on SNMP for dynamic information. *Journal of Network and Computer Applications, 30*(1), 331–353.

Tokihiro, F., Masayuki, H., & Takuji, K. (2006). Agent system for operating web-based sensor nodes via the Internet. *Journal of Robotics and Mechatronics, 18*(2), 186–194.

Tou, C. J., Lin, C. M., & Lin, J. M. (2006). *Network agents*. Taipei, Taiwan: Acore Book Corporation.

Trifan, M., Ionescu, B., Ionescu, D., Prostean, O., & Prostean, G. (2008). An ontology based approach to intelligent data mining for environmental virtual warehouses of sensor data. In *Proceedings of IEEE conference on virtual environments, human–computer interfaces and measurement systems, Istanbul, Turkey* (pp. 125–129).

Tseng, C. S., & Liu, T. L. (2007). Developing a monitor platform for high-quality academic research in Taiwan. In *Proceedings of 2007 Taiwan network conference, Taipei, Taiwan*.

Tzou, W. S., Lee, W. O., & Jeng, B. C. (2009). Exploiting knowledge ontology and software agents for PPI network analysis. *Expert Systems with Applications, 36*(10), 12605–12612.

Valliyammai, C., & Selvi, S. T. (2008). Relational network monitoring system for grid performance optimization. In *Proceedings of the sixteenth international conference on advanced computing and communications, Chennai, India* (pp. 170–173).

WinPcap (2009). *The Windows packet capture library*. Available at <http://www.winpcap.org/> (visited on Nov. 5th, 2009).

Wu, F., Zhao, Z., & Ye, X. (2008). A new dynamic network monitoring based on IA. In *Proceedings of 2008 international symposium on computer science and computational technology, Shanghai, China* (pp. 637–640).

Wu, J., Yuan, S. F., Ji, S., Zhou, G. Y., Wang, Y., & Wang, Z. L. (2009). Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring. *Expert Systems with Applications*.

Yang, C. Y. (2004). *IIS 6 Server setup and management*. Taipei, Taiwan: Kings Information Book Corporation.

Yang, H. M. (2007). *Developing a network management system based on Web*. Harbin, China: Science and Technology Innovation Herald, China Astronautic Publishing House.

Yang, S. C., & Tseng, L. M. (2007). Flooding detection and notification system over aggregate network. In *Proceedings of 2007 Taiwan network conference, Taipei, Taiwan*.

Yang, S. Y., & Lu, S. H. (2009). A study on ontology-supported information recommendation system for scholars. In *Proceedings of 2009 international conference on advanced information technology, Taichung, Taiwan*.

Yang, F. J., & Yang, F. R. (2007). *Introduction to TCP/IP*. Taipei, Taiwan: XBook Corporation.

Yoshida, K., Shomura, Y., & Watanabe, Y. (2007). Visualizing network status. In *Proceedings of 2007 international conference on machine learning and cybernetics, Hong-Kong, China* (pp. 2094–2099).