

# Accounting management system enhancement supporting automated monitoring and storing facilities

C. Bouras S. Kastaniotis

<sup>1</sup>Computer Engineering and Informatics Department

University of Patras, Greece

<sup>2</sup>Computer Technology Institute, Greece

e-mails: bouras@cti.gr,kastansp@cti.gr,

## Abstract

In this work we present the design, development and implementation of a set of tools embedded in a network management system used in the University of Patras Network (UPatrasnet), providing advanced accounting management services. These tools help the critical network parameters monitoring and storing procedure in order to provide the information needed for efficient management. We defined which resources to be watched and what kind of information to be collected by the management system in order to deploy the specifications of the processing and analyzing tools. Finally, we used HP OpenView as a network manager.

## Keywords

Accounting management, MIB, SNMP, Network resources, Network administration

## 1. Introduction

All current management architectures have been developed after the design of network functions had been completed. Such approach indicates a specific conceptual view on the role of management functions and invites to apply different architectural concepts for the design of management functions.

Users obviously want the best possible network bandwidth / performance at the lowest possible price. On the other hand, it is unrealistic to expect that it will always be possible to find a good pricing and billing method for each network during the phase of designing the management functions.

In order additional experience to be obtained during the operational phase of a network there must be an enhanced subsystem supporting accounting management, e.g., the gathering and recording of data related to the network resources usage.

This work presents a set of applications that completes the accounting facilities offered by HP OpenView network platform. These applications help in monitoring the traffic for an administrative domain (UPatrasnet), as well as the storing of the formatted accounting information in a database.

Statistical (accounting) data is used for the right administrative decisions to be taken in order major network problems (congestions, drops, resource insufficiencies) to be avoided. Furthermore, the statistical data will be used as an input to a future charging and pricing policy, so that the appropriate network resources to be allocated with a fair and convenient way.

All the applications developed offer secure and easy administration.

This paper is organized as follows: in section 2 there is a short description of the applications context. The third section provides details of the applications architecture, specifically presenting analysis, design and implementation issues. In section 4 the functionality offered by the applications is outlined. The last section summarizes conclusions elicited from our work and presents future work issues.

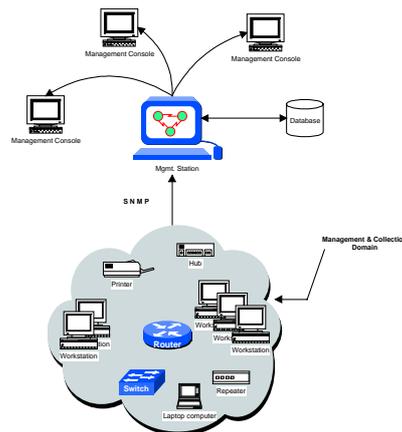
## 2. Description of the applications

### 2.1 General Overview

The applications developed cover a wide range of accounting issues. The traffic information involves information about the flows that pass through a router, information about the selected routers, information about performance and congestion, information about the selected items (hosts) and information about all the events and traps (these that have been configured) generated during network operation.

We can distinguish the application in three categories. The first category provides traffic information from the network of UPatrasnet to Internet and vice versa, the second one for a specified node of UPatrasnet and the last one for a specified subnetwork of UPatrasnet (or a set of subnetworks of UPatrasnet).

### 2.2 Infrastructure



**Figure 1: The management model on which the applications are implemented**

The set of applications was designed, developed and implemented on a fully centralized network management model in which the entire network comprises a single management domain (UPatrasnet) and one powerful management station performs all monitoring and data collection for that domain (Figure 1).

## 3. Architecture

### 3.1 Applications Analysis

The design of the set of applications was based on a user needs analysis conducted, using interviews with the subnetworks administrators of UPatrasnet and questionnaires. This survey led to the following functional specifications:

- There should be a mechanism to access the routers without explicitly using telnet, so we used a simple-minded program to talk to the cisco console.
- There must be a Relational Database Management System (RDBMS) that works behind and provides the whole system with an efficient way of manipulation. The RDBMS will keep all the statistical data needed in order after their processing the appropriate decisions to be taken.
- All applications should provide a graphical user interface (GUI) to easily make anyone familiar with them. One way to achieve this is to access the applications through the OpenView Windows environment.
- Administrator must focus on managing the most critical elements and processes of the network, and build complete solutions for them.
- The menu items of the implemented applications should be displayed only on specific submaps, not on all maps of the OpenView Windows (OVW) environment.
- Due to the fact that UPatrasnet is a heterogenous network there are a lot of devices (entities) that does not support the SNMP protocol. There must be a way of distinguishing these *unmanaged* objects on the map.

There are, also, some non-functional specifications which ensure the quality of the accounting applications. They fall into the following categories.

#### *Performance*

- 24 hours availability of the applications.

#### *Safety*

- Any application must be able to recover from a failure within a reasonable period of time and without affecting other applications.
- Usage of backup mechanism in order to ensure the integrity of data.

#### *Security*

- The consistency of the data must be ensured.

#### *Maintenance*

- Small changes of enhancement, extension or adaptation can be performed without necessitating redevelopment of the applications.
- The applications architecture must be open in terms of smooth integration with new products or new services as additional experience is gained.

#### *Environment*

- The possible administrator must be able to use the applications' facilities within a short period of time.

### **3.2 Applications design**

To determine which variables must be watched, administrators must understand the precise meaning of many management issues.

We have adopted the accounting framework used by OSI, which defines a generalized accounting management activity, that includes calculations, usage reporting and enforcing various limits on the use of resources.

The OSI accounting model defines three basic entities: 1) the *METER*, which performs measurements and aggregates the results of those measurements, 2) the *COLLECTOR*, which is responsible for the integrity and security of *METER* data in short-term storage and transit, and 3) the *APPLICATION*, which processes/formats/stores *METER* data. *APPLICATIONS* implicitly manage *METERS* [RFC 1272 (1991)].

The accounting model can be described as a complex client / server model that operates in two levels. In the first level the *METER* is the server, while the *COLLECTOR* is the client. In the second level, the *COLLECTOR* is the server, while the *APPLICATION* is the client. Obviously, the *COLLECTOR* plays a dual role in the model, that of a client and a server.

We recommend SNMP as the basic communication framework and more precisely a secure version of SNMP, such as SNMPv2.

What we did is the development of a set of applications (*Traffic Monitoring*) making use of an accounting MIB (*MyAccMIB*) written in ASN.1 notation. The applications make use of the OVW NNM interface, under selected menu items.

The communication between an accounting agent and meters or between an accounting manager and accounting agents achieved through trap-directed polling. In each of these cases, administrator uses polling to periodically obtain an update of the accounting data from the routers.

OSI	TCP/IP protocol suite
Application	<i>Traffic Monitoring</i>
Presentation	ASN.1 ( <i>MyAccMIB</i> )
Session	
Transport	Host-to-Host
Network	Internet
Data Link	Network Access
Physical	

**Figure 2: The applications hierarchy in the TCP/IP protocol stack**

The information exchange with all the routers is done using globally the same community string “*public*”. It is because we want to communicate with routers without using telnet and making use of the same mechanism for all of them.

Figure 2 illustrates the hierarchy view of our implementation as far as ASN.1 notation and ‘Traffic Monitoring’ concerns.

### 3.3 Implementation Issues

The use of *ANSI C* language is considered to be as an adequate choice for writing short and smart programs that can be easily modified in order to make

several tests before the final implementation. Thereby, little work is needed for modifying the existed code so as new applications and services to be built.

*Awk* processing language has the unique capability of providing filters, entirely necessary for isolating the useful information. Using *awk* we managed to modulate the ip accounting table taken from several routers of UPatrasnet and finally obtain the desired information.

*ASN.1* is a language for describing structured information; typically, information intended to be conveyed across some interface or communication medium. It is widely used in the specification of communication protocols.

With *ASN.1*, the protocol designer can view and describe the relevant information and its structure at a high level and need not be unduly concerned with how it is represented while in transit. Compilers can provide run-time code to convert an instance of user or protocol information to bits on the line, using BERs (Basic Encoding Rules).

Finally, *ASN.1* defines a set of simple types and specify them with a tag. We can refer to these types using this tag. Beyond the definition of types, *ASN.1* provides a symbolism for the definition of the values of the above types. Subsequently, it defines mechanisms for constructing more complex types and tags for these types.

We have expanded an SNMP agent with the proposed Accounting MIB written in *ASN.1*.

#### *Symbol Registration Files (SRFs)*

HP OpenView NNM provides a whole range of Symbol classes and subclasses. The range of symbols can be enriched by means of SRFs.

Using the facility of SRFs we managed to assign custom symbols to specified nodes with a particular `sysObjectID`. Network Node Manager uses a file (`oid_to_sym`) to map an object's `sysObjectID` to the default *ovw* symbol type used to represent the node in the IP topology submaps it displays. Taking this fact into account we can cause all non-SNMP nodes to be marked unmanaged when automatically discovered. The special value that must be substituted for the SNMP object id (`sysObjectID`) for unmanaged nodes is `DEFAULT_IP`.

#### *Application Registration Files*

Application Registration Files (ARFs) have been used to integrate Network and System Management Applications with the *OVW* user interface. Many aspects of the application's interaction with *OVW* are defined using ARFs.

The Application Registration File contains fields that define the menu bar items, tool bar items and symbol pop-up items from which the application will be accessed, and fields that specify the command line, which invokes the application.

In more detail, applications are integrated into *OVW* by using structures in the ARFs called Application Blocks. Application Block defines many aspects of the interfaces between *OVW* and application and for a single application

specifies the name of the application, the version of the application, the description of the application, the copyright information for the application, how the application is incorporated into the OVW Menu Bar, selection rules for the application and instructions for invoking the application command.

Application integration begins in the main menu bar. A MenuBar block is specifies the menu label on the menu bar under which the application will eventually appear. An optional mnemonic character can be specified to allow keyboard selection by typing a single character.

Additional menu items and the application invocation are defined through a Menu statement within the MenuBar block. The Menu statement specifies the menu label and its function.

```
MenuBar <100> "Misc" _i
{
  <100> "Traffic Monitoring" _T f.menu "Traffic Monitoring";
}

Menu "Traffic Monitoring"
{
  <100> "General Traffic" _G CONTEXT "AllContexts || isIP"
  f.menu "General Traffic";
  <100> "Node Traffic" _N CONTEXT "AllContexts || isIP"
  f.menu "Node Traffic";
  <100> "Subnet Traffic" _S CONTEXT "AllContexts || isIP"
  f.menu "Subnet Traffic";
}

Menu "General Traffic"
{
  <100> "Internet->150.140.0.0" _I CONTEXT "AllContexts || isIP"
  f.action "General From Internet";
  <100> "150.140.0.0->Internet" _I CONTEXT "AllContexts || isIP"
  f.action "General To Internet";
}

Menu "Node Traffic"
{
  .....
```

**Figure 3: A sample of the code generates the menu items of the applications**

### *Facilities of HP OpenView Network Node Manager*

The administrative platform OVW provides a complete Graphical User Interface (GUI), which helps the creation of ARF files. For instance, using point-and-click, we could choose whether an application should display data in the form of a graph, table or form. The GUI also lets the developer key in the appropriate help text that should be linked with the application.

## **4. Functionality**

All applications developed support point-and-click operations on the map of OVW NNM integrated environment. In proportion with the application used, the network administrator can select one active network device (e.g., a router) and several nodes from the map where the applications are active, and run the desired application. Obviously, multiple selections are allowed (multiple nodes can be selected each time by using *ctrl-click*).

If the network administrator wants to get information for a specific node on the map and cannot find it by the first look, he can use a dialog box to insert its ip address or alias name.

For the case a subnetwork is involved, only the dialog box can be used in order the specified subnetwork to be selected by the administrator.

The implemented applications capture only specific traffic information (specified by MyAccMIB variables) so that to indicate critical data and states of the whole network.

We made the assumption that there is a network administrator to whom Internet accounting is of interest. He "owns" and operates some subset of the Internet (one or more connected networks) that may be called his "administrative domain". This administrative domain has well defined boundaries. The network administrator is interested in 1) traffic within his boundaries and 2) traffic crossing his boundaries. Within his boundaries he may be interested in end-system to end-system accounting or accounting at coarser granularities (e.g., university department to university department).

The developed applications support three categories of traffic information. The first category provides traffic information from the network of University of Patras (UPatrasnet) to Internet and vice versa. This category includes the following modules:

- Traffic monitoring from Internet to UPatrasnet.
- Traffic monitoring from UPatrasnet to Internet.

The second category provides traffic information for a specified node of UPatrasnet. This category includes the following modules:

- Traffic monitoring from a node of UPatrasnet to UPatrasnet.
- Traffic monitoring from a node of UPatrasnet to Internet.
- Traffic monitoring from UPatrasnet to a node of UPatrasnet.
- Traffic monitoring from Internet to a node of UPatrasnet.

The third category provides traffic information for a specified subnetwork of UPatrasnet (or a set of subnetworks of UPatrasnet). This category includes the following modules:

- Traffic monitoring from a subnetwork of UPatrasnet to UPatrasnet.
- Traffic monitoring from a subnetwork of UPatrasnet to Internet.
- Traffic monitoring from UPatrasnet to a subnetwork of UPatrasnet.
- Traffic monitoring from Internet to a subnetwork of UPatrasnet.

#### *Accounting MIB*

The Accounting MIB developed is subdivided into six groups:

- **Flow:** information about the flows that pass through the selected router.
- **Meter:** information about the selected router on the map and its interfaces gathered by meters.
- **Congestion:** information about subnetwork performance and congestion.
- **Host:** information about the selected host on the map.
- **Event:** a table of all events generated by the SNMP agent.

In all cases we managed to export accounting data from the selected router into text files, formatting them similarly to the accounting MIB MyAccMIB.

All statistical (accounting) data are stored automatically (after monitoring) in an Oracle database and are kept there for a time period not longer than three months. Each saved file is been stamped by the date and time of creation and the selections made on the map, before using a specific application module.

## 5. Conclusions and future work

Working for the UPatrasnet we implemented a set of applications enhancing the existing accounting facilities supported by HP OpenView platform. We defined an Accounting MIB (MyAccMIB) and used HP OpenView as an Accounting Manager. We recommend SNMP as the basic communication framework and more precisely a secure version of SNMP, such as SNMPv2. Our approach is completed by a storing mechanism for saving formatted accounting data into a database.

We can expand the current accounting applications into an integrated accounting system included pricing policies. Pricing computer network traffic will be essential to manage network traffic and to control and provide different QoS required by different classes of applications and users.

For the time being, there are two pilot projects, MBS (Managed Bandwidth Services) and TrACe (Traffic Analysis and Charging Pilot Project), which involve traffic measurement, analysis, charging and bandwidth management. GUNet (Greek Universities Network, [www.gunet.gr](http://www.gunet.gr)) funds both projects and the results of the completed work will be announced in the near future.

## 6. References

- Stallings W. (1996), *SNMP, SNMPv2 and RMON, Practical Network Management*, Second Edition, Addison-Wesley.
- Apostolopoulos T.K., Prokos I.G. (1998), "ACCOUNTING MANAGEMENT IN TCP/IP NETWORKS: CONCEPTS, ARCHITECTURE AND IMPLEMENTATION", *Proceedings of the Second IASTED International Conference PARALLEL AND DISTRIBUTED COMPUTING AND NETWORKS*, Brisbane, Australia.
- RFC 1272 (1991), *Internet Accounting: Background*.
- Fang K., Leinward A. (1993), *Network Management: A Practical Perspective*, Addison-Wesley.
- Stevens W. R. (1991), *Unix Network Programming*, Prentice Hall.
- Cisco Systems (1994), *Cisco MIB variables for version 10.0 of system software*.
- Jander M. (1993), *Router Management Goes Beyond Pretty Pictures*, Data Communications.
- HP OpenView Network Node Manager, *Fundamentals for Network Managers, Student Workbook*, Version C.01.
- HP OpenView (1997), *Using Network Node Manager*, Edition 1.
- Stevenson D. W. (1995), *Network Management: What it is and what it isn't*, Network Management White Paper.