19

A quota system for fair share of network resources

Çelik C.
Computer Center
Middle East Technical University
Inonu Bulvari, 06531
Ankara, Turkiye
can@knidos.cc.metu.edu.tr

Özgit A.
Dept. of Computer Engineering
Middle East Technical University
Inonu Bulvari, 06531
Ankara, Turkiye
ozgit@metu.edu.tr

Abstract

Interconnected networks of today provide a wide variety of services, which consume widely differing amounts of resources. But unlike other computing resources such as disk space and processing power, the network resource is not that much accounted.

Internet Engineering Task Force (IETF) internet-accounting working group is currently studying this subject. Their approach to the problem is focused on network accounting but does not cover any real-time controls such as quotas or enforcement.

In this paper, a model that increases coordination between accounting mechanisms and access controls is introduced. This model is compatible with the concepts and the architecture introduced by IETF internet-accounting working group. In the proposed model the quota manager is responsible from producing a table of service consumers that have already reached their quotas. This table is formed by using the data accumulated by the accounting system.

Keywords

Network Management, Network Accounting, Quota System, TCP/IP, SNMP.

A. S. Sethi et al. (eds.), *Integrated Network Management IV*

1 INTRODUCTION

Today computer networks have become a fundamental part of computing. They are used for serving many purposes such as file transferring between computers, cross-login connections, file sharing, distributed computing, electronic mailing, electronic discussion lists, information services, etc. Since the 'network' as a shared physical resource is limited for most cases, it is a reasonable approach to account the usage of network bandwidth. It could also be necessary to impose limitations for the usage, in order to prevent network misuse or even abuse.

This paper is based on the work being carried out by the IETF internet-accounting working group. It describes a system that uses IETF working group's accounting model and adds a quota system to it.

The Internet-accounting architecture model proposes a *meter* that listens on the network to collect information about network usage (Mills, 1991) (Mills, 1992) (Brooks, 1993). A *network manager* tells the meter what kind of information is needed and how much detail the accounting data should contain. This paper introduces a quota system which uses the data collected by the meter and forms a list of hosts that have already reached their quotas. Each service provider such as gateways, file servers, compute servers, etc., may check this list before they serve their users. If a service provider encounters any host that is in the list, it may refuse to provide any service to that host.

After a discussion of the milestones of Internet Accounting Architecture in Section-2, IETF's Internet Accounting Architecture is described in Section-3. The first implementation of the architecture is presented in Section-4. In Section-5, the proposed quota architecture is discussed.

2 HISTORY OF INTERNET ACCOUNTING

IETF Internet Accounting Working Group was formed with the goal to produce standards for the generation of accounting data within the Internet that can be used to support a wide range of management and cost allocation policies. The first publication of the group was titled as 'Internet Accounting Background RFC-1272', published in November 1991 (Mills, 1991).

The milestones of the working group are the following :

- Internet-accounting Background RFC-1272 was published (Mills, 1991).
- SNMP was recommended as the collection protocol.
- Internet-accounting architecture was submitted as Internet-Draft (Mills, 1992).
- Internet-accounting meter MIB was submitted as Internet-Draft (Brooks, 1993).
- The two drafts mentioned above expired 6 months after submission as a draft. And they were then modified several times by the working group.
- Internet-accounting working group was suspended in April 1993, waiting for feedback from implementation experience.
- The first implementation came in October 1993; NeTraMet & NeMaC (Brownlee, 1993).
- The working group started again in March 30 1994. They are planning to publish 'Internet Accounting Architecture' and 'Internet Accounting MIB' RFCs.

3 INTERNET ACCOUNTING ARCHITECTURE

The Internet accounting model, currently a draft of a working group (Mills, 1992), draws from the OSI accounting model. It separates accounting functions into the parts shown in Figure 1.

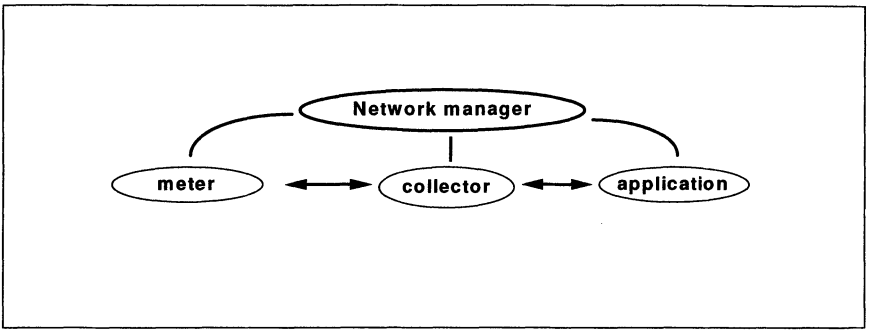

Figure 1 Internet Accounting Functions.

- Network Manager (or simply, Manager) : The network manager is responsible for the control of the meter. It determines and identifies backup collectors and managers as required.
- Meter : The meter performs the measurement of network usage and aggregates the results.
- Collector : The collector is responsible for the integrity and security of data during transport from the meter to the application. This responsibility includes accurate and preferably unforgeable recording of accountable (billable) party identity.
- Application : The application manipulates the usage data in accordance with a policy, and determines the need for information from the metering devices.

The data exchange can be categorized as follows:

- Between Meter and Collector
  The data which travels this path is the usage record itself. The purpose of all the other exchanges is to manage the proper execution of data exchange.
- Between Manager and Meter
  The manager is responsible for controlling the meter. Meter management involves commands which start/stop usage reporting, manage the exchange between meter and collector(s) (to whom do meters report the data they collect), set reporting intervals and timers, and set reporting granularities. Although most of the control information consists of commands to the meter, the meter may need to inform the manager of unanticipated conditions as well as responding to time-critical situations, such as buffer overflows.

- Between Manager and Collector
  Feedback on collection performance and controlling access to the collected traffic statistics are the main reason of this traffic. In most implementations, the manager and thecollector will be the same entity.

Since redundant reporting may be used in order to increase the reliability of usage data, exchanges among multiple entities are also considered, such as multiple meters or multiple collectors or multiple managers.

Internet accounting architecture assumes that there is a "network administrator" or "network administration" to whom network accounting is of interest. The administrator owns and operates some subset of the internet (one or more connected networks) that may be called as "administrative domain". This administrative domain has well defined boundaries. The network administrator is interested in (i) traffic within domain boundaries and (ii) traffic crossing domain boundaries. The network administrator is usually not interested in accounting for end-systems outside his administrative domain (Mills, 1991).

SNMP is the recommended collection protocol. A draft SNMP MIB is already proposed (Brooks, 1993).

The following points are not covered by the IETF working group's proposal :

- User-level reporting is not addressed in this architecture, as it requires the addition of an IP option to identify the user. However, the addition of a user-id as an entity at a later date is not precluded by this architecture.
- The proposal does not cover enforcement of quotas at this time. A complete implementation of quotas may involve realtime distributed interactions between meters, the quota system, and access control.

In the following sections of the paper, a model is introduced which will add a quota system to IETF's proposed architecture.

4 THE FIRST IMPLEMENTATION OF THE PROPOSED INTERNET ACCOUNTING ARCHITECTURE (NeTraMet & NeMaC)

The first implementation of the Internet accounting architecture is NeTraMet (Network Traffic Meter) and NeMaC (NeTraMet Manager/Collector) (Brownlee, 1993).

In this implementation, *network manager* and *collector* are the same entity. *Meter* is another software which can be located on the same host with *manager/collector* or can be located on a different host.

A *traffic flow* is a stream of packets exchanged between two network hosts. *Manager/collector* sends a set of rules to the *meter* which are used for deciding which flows are to be considered and how much detail about the flow will be collected. Rules can be quite detailed so that one can define flow of specific protocols. For example such rules can be stated.

'Count those packets from host X to host Y that are in TCP protocol' or
'Count those packets transferred via telnet connections'

Rules are sent from *manager/collector* to *meter* in SNMP format. Actually, they are variables set in the MIB located in the *meter*.

Figure 2 shows the traffic between *meter* and *manager/collector*.

The *meter* starts collecting data, considering the rules received from *manager/collector*. The flow data collected from the network is also put in the MIB-accounting database that is located in the meter, and the collector gets this data at regular time intervals. All the communication between *manager/collector* and *meter* is done via SNMP.

The MIB for Internet accounting is located in the meter. The structure of this MIB is explained in the following paragraphs.

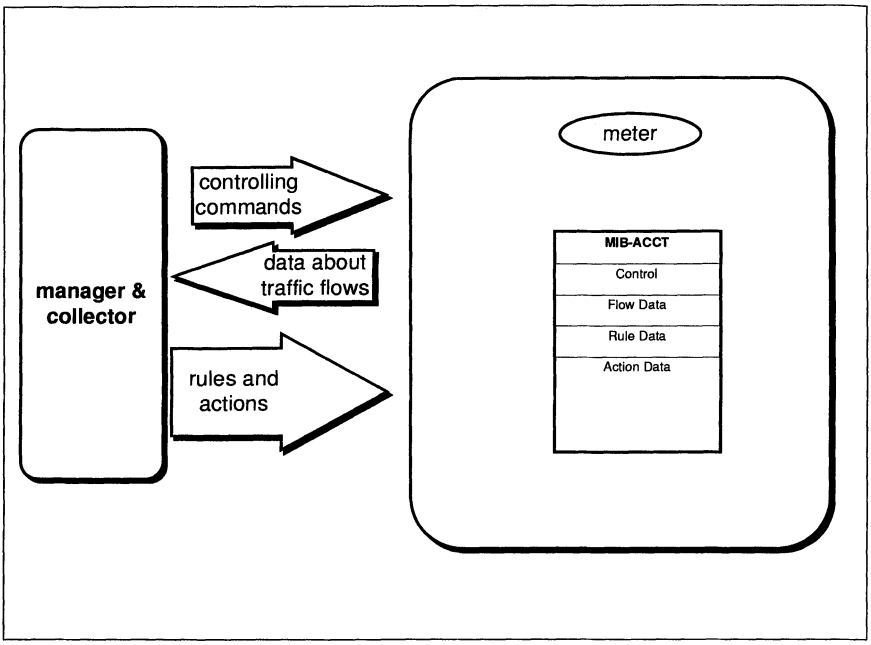

Figure 2 Manager/Collector and Meter.

MIB-acct is composed of four major parts.

- Control : Some parameters to control the meter such as sampling rate, when to send a trap to the manager if the meter is running out of memory, etc.
- Flow data : The counted flows are put here.
- Rule data : Rules for deciding if a flow is to be considered.
- Action data : Action to be performed if the rule's value is matched such as count, tally, aggregate.

'NeTraMet & NeMaC' is the first implementation of internet accounting architecture. NeTraMeT also implements the internet accounting meter services MIB. NeTraMet is

available under SunOS and MSDOS operating systems. NeMaC is available under SunOS, SGI-IRIX and HP_UX operating systems. The quota system described in the next section is implemented by using some parts of this software.

5 A QUOTA SYSTEM FOR INTERNET ACCOUNTING ARCHITECTURE

The quota system proposed in this paper is an extension to the IETF's proposed internet accounting architecture.

5.1 Architecture

The accounting system described in 'Internet Accounting Architecture' section collects the accounting data. Quota system processes this data in order to form a list of hosts that have used the system resources beyond their quotas. This list is called the black-list. The algorithm used for deciding which hosts will stay in the black-list and how long will be described in the 'Algorithm' section. The black-list is valid in some domain in the internet. This domain is the mapping of the 'administrative domain' of the 'Internet Accounting Architecture'. More than one copy of the black-list can be located in a domain.

The black-list has been implemented as a MIB entry that can be located on any host running SNMP. It is actually an array of IP addresses. In order for implementing the quota system standard MIB has been modified by adding new variables. The added MIB variables in the ASN.1 notation are shown in Table 1:

Table 1 MIB-quota

```
blacklist OBJECT IDENTIFIER ::= { experimental 100 }

blacklistTable OBJECT-TYPE
        SYNTAX SEQUENCE OF blacklistEntry
          ACCESS read-write
          STATUS mandatory
          ::= { blacklist 1 }

blacklistEntry OBJECT-TYPE
          SYNTAX  IpAddress
          ACCESS  read-write
          STATUS  mandatory
          ::= { blacklistTable 1 }

NoOfEntry OBJECT-TYPE
          SYNTAX  INTEGER
          ACCESS  read-write
          STATUS  mandatory
          ::= { blacklist 2 }
```

The first entry 'blacklist' is the highest entry in the MIB-blacklist hierarchy. Its long name is 'iso.org.dod.internet.experimantal.blacklist'. This variable does not hold any value.

The 'blacklistTable' MIB variable defines an array of 'blacklistEntry'. Each 'blacklistEntry' holds an IP address and it is indexed by those addresses. The 'NoOfEntry' shows the number of hosts in the MIB blacklistTable. To set this variable to 0 clears the blacklistTable.

Quota manager has been implemented as a part of *network manager* software. It fills the black-list, the MIB-quota, by using SNMP. MIB quota is a dynamic list and the quota manager decides which IP addresses will enter to and which will exit from the list. Quota manager is also responsible from the consistency of the black-lists, if more than one of them are located in the domain. *Quota manager* does this by updating all of the black-list servers whenever an update is needed.

Service providers (gateways, FTP servers, NFS servers, etc.) in the domain may check the black-list before providing any service and do not give any service if the service requesting host is in the black-list. Each service provider knows which host(s) has up to date black-list in their MIBs and by using SNMP it checks if the service requester is in the black-list or not.

Since the 'Internet Accounting Architecture' allows more than one meter per a Network Manager, the network and quota mangers can use information coming from different networks in the domain.

Figure 3 shows the simplest configuration of the quota system with one meter, one black-list (MIB quota), and one network.

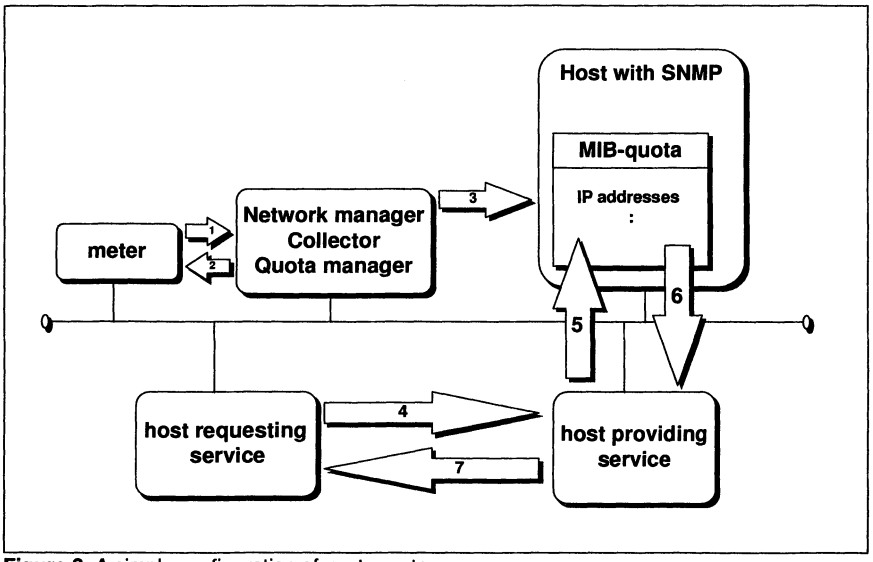

Figure 3 A simple configuration of quota system.

In Figure 3, the arrows denote interactions among various entities. These interactions are explained below.

1,2 : The communication between *meter* and *manager/collector*. This communication is the same as the first implementation of internet-accounting architecture in which *manager* controls the *meter* and *meter* sends the usage reports to the collector.

3 : The *quota manager* fills the MIB-quota on regular time intervals.

4 : A service is requested, such as an ftp request.

5 : The service provider, namely the ftp server, checks if the service requester is in the black-list or not. This is achieved by an SNMP session between *quota manager* and the service provider. It is actually an SNMP-get request of the MIB variable 'iso.org.dod.internet.experimental.backlist.blacklistTable.blacklistEntry.IPaddress made by the service provider.

6 : The answer comes from the host running SNMP agent that maintains MIB-quota. If the IP address of the host requesting the service is in the MIB-quota, it returns that IP address, otherwise it returns something like 'No Such Variable'.

7 : If the IP address of the host requesting service is returned from black-list server, the service provider may not provide the service as the requester is found in the black-list and may return an error. This part is purely implementation dependent. Administrator could implement various alternative models depending on the policy set for that domain.

The proposed quota system can use multiple copies of MIB-quota in a domain. This provides two advantages :

- Availability : If a problem occurs in one of the black-list servers, the alternative one still can be accessed. Of course each service provider knows all of the black-list servers in the domain. They have to know which black-list server to contact first, which one to contact next and so on. Although the updating times of the black-list servers may differ, this won't be a big problem since they are being filled by the same quota-manager.
- Access speed : If the domain is formed of multiple networks, then there will be a performance problem for the service providers to check the black-list through gateways. In such cases, a black-list server can be configured for each network in the domain.

If multiple copies of MIB-quota are desired, the quota manager makes the updates to all of the copies. The updates will be done on regular time intervals. These intervals can be tuned either statically or dynamically by considering the load on the network.

Figure 4 shows a more complicated configuration in which there are three networks, three meters and two black-list servers (MIB-quota).

In the figure the symbols stand for :

R : The router between 3 networks.

Black arrows : The traffic between meters and Manager/Collector. Each of the three networks has a meter in this configuration. Each meter reports the network usage information to the collector. And each of them is controlled by the manager.

X labeled arrows : Since there are two black-list servers in this configuration, the quota manager needs to update both of them on regular time intervals. In order to make necessary additions and deletions to/from MIB-quota, the quota manager makes SNMP-set requests to blacklist-servers. These requests are the same for both of the blacklist-servers.

A labeled arrows : A service is requested from a service provider.

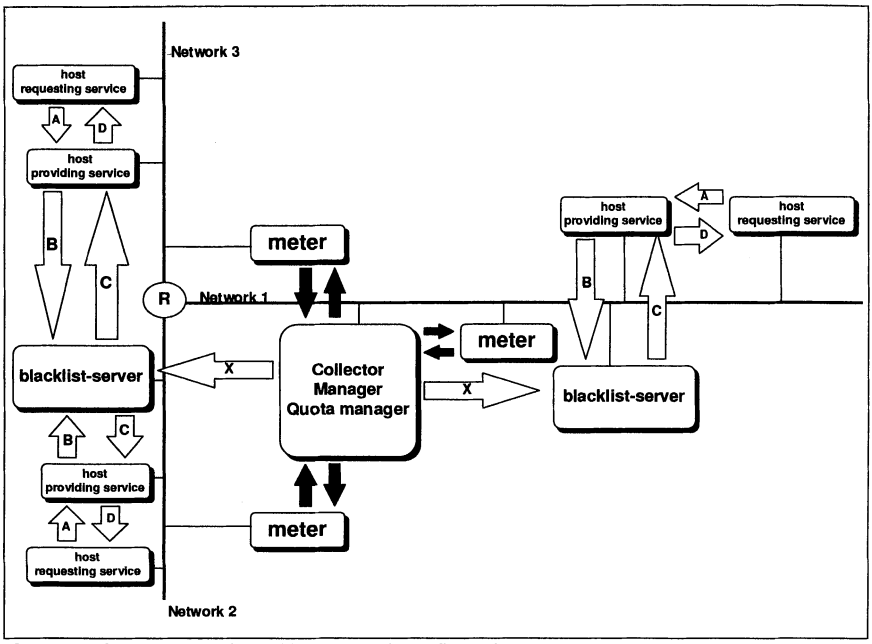

Figure 4 Quota system in multiple network configuration.

B labeled arrows : The service provider checks if the service requester is in the black-list or not. This is implemented by an SNMP-get request. Each service provider makes this request to the nearest blacklist-server. The servers on Networks 1 and 2 makes this request to the blacklist-server on the same network. The one on network 3 makes this request to the blacklist-server on the Network2.

C labeled arrows : This is the answer coming from the blacklist-server. If the IP address of the host requesting the service is in the MIB-quota, blacklist-server

returns that IP address, otherwise it returns something like 'No Such Variable'.

D labeled arrows : If the address is in the black-list. The service provider may or may not provide the service. It returns either an error message or the normal response message depending on the specific implementation.

5.2 Algorithm

This algorithm decides which hosts will be put in the black-list and how long they will stay there. Each host starts with U variable assigned to 0 which indicates that no network resources have yet been used. Whenever the host uses the network, the U variable increases proportional to the network usage until a limit *HIGH* is reached. That time the host enters the black-list.

Every night another part of the software decreases the U variable by D. D is the daily increment to the quota of the host. This gives extra network usage chance to the host. A host in the black-list can not use the network resources authenticated by the quota manager. But every night its U variable is decreased. If that comes down to *LOW*, the host is deleted from the black-list. In the current implementation by default U is decreased every night, however this interval can be changed by the network administrator. The network administrator can even give extra usage chance to some of the hosts without considering the algorithm. Another approach can be charging the users for decreasing their U variable and have them to use extra resources.

This is a dynamic quota mechanism, if the host does not use the network its quota is increased, but to some limit. And if it uses the network its quota decreases and enters the black-list if the usage is higher than allowed.

The following figures (Figure 5 and Figure 6) describe the algorithm in flowchart form.

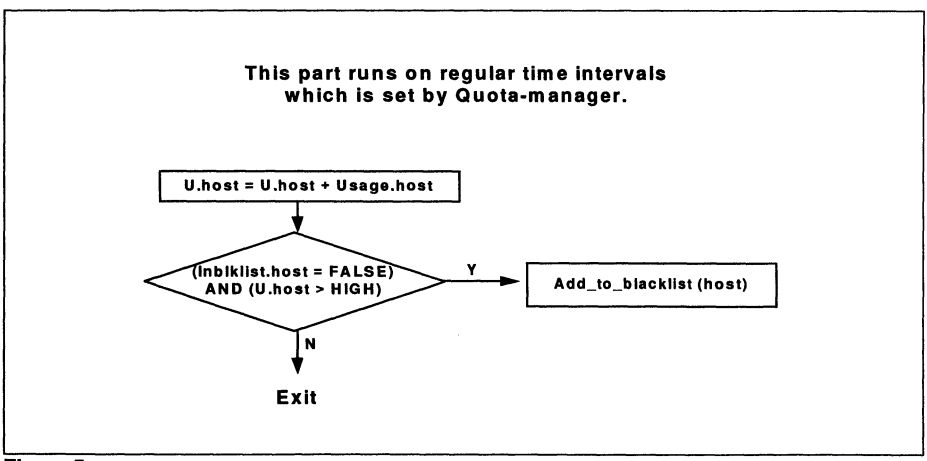

Figure 5

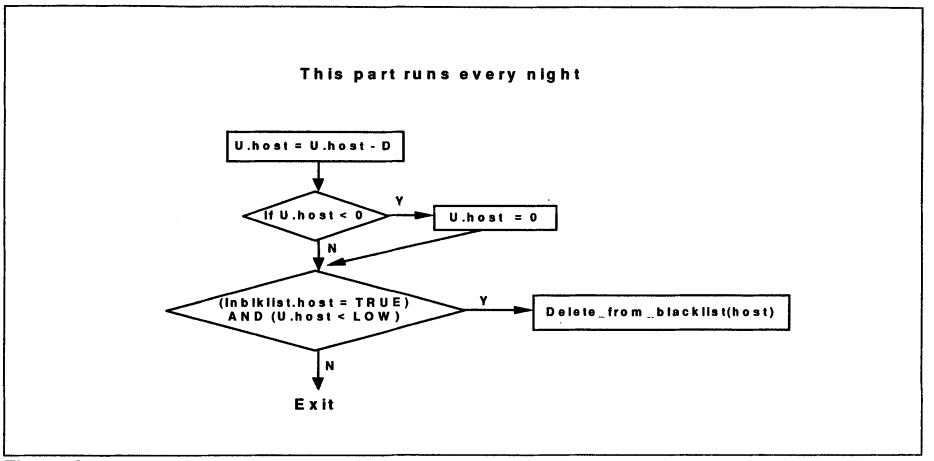

Figure 6

Inblklist.host is TRUE if the host is in the black-list, Add_to_blacklist(host) adds the host to the black-list, Delete_from_blacklist(host) deleted the host from the black-list.

6 SUMMARY

Problems arising from highly loaded networks are not unusual today. Any available resource is consumed by users in a short time. Increasing the available bandwidth does not guarantee to solve this problem permanently. There seems to be a lack of tools that provide a fair share of network resources such as bandwidth. In this study a quota system is proposed to solve this problem in local environments.

Since TCP/IP is the most common networking protocol and SNMP is the most common network management protocol, the study is based on these protocols. As a result of this, it can be ported to many platforms. With the help of this system, network managers may put usage limitations on some of the resources. And this provides a fair share of these resources.

The architecture proposed in this paper could be applied to service usage other than just bandwidth. Meter can collect the traffic for any specific protocol and quota manager can use this data for deciding the usage. A combination of protocols can also be used for deciding the usage.

7 REFERENCES

Brooks, C. (1993) Internet draft. Internet accounting: MIB.

Brownlee, N. (1993) Introductory documentation NeTraMeT & NeMaC (Network Traffic Meter & NeTraMeT Manager/Collector).

Mills, C. Hirsh, D. Ruth, G. (1991) RFC 1272 Internet accounting: background.

Mills, C. Laube, K. Ruth, G. (1992) Internet draft. Internet accounting: Usage Reporting Architecture.

8 BIBLIOGRAPHY

Can Çelik has graduated from Computer Engineering department of Middle East Technical University (METU) in 1991. He is a graduate student in the Computer Engineering Department of METU, he is expected to get M.Sc. degree in Jan. 1995. Mr. Çelik is doing Systems Programming in the Computer Center of METU, specialized in UNIX operating systems.

Attila Özgit is a graduate of Middle East Technical University. He is a faculty member of the Computer Engineering Department and also the Director of Computer Center. His research interests are Operating Systems, Computer Networks and Distributed Systems.