Implementation of Charging Schemes to Transport and Service Level ATM Networks

By Yining Chen, B Eng

Supervisor Dr T Curran

Submitted for

M Eng in Electronic Engineering

Dublin City University

Submitted May 1998

Table of Contents

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of M Eng in Electronic Engineering is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work

Signed ____________________ Date ____________________

Candidate ID No ____________________

Acknowledgements

I would like to thank Dr T Curran for his encouragement and help Acknowledgement is also due to Fionna Lodge, Jimmy McGibney and everyone in my lab for proof-reading this thesis

Additionally, this thesis would not have been possible without the help and encouragement of my wife, Meng Not only was she able to stay awake while reading the manuscript, but she also gave me her complete support from the beginning to the end

Chapter 1. Introduction

1 1 Introduction

Nowadays, telecommunications networks like telephony networks, computer networks, and packet switched networks are all dedicated to only one or just a few types of services When a user wants to subscribe to various telecommunications services, he needs to be connected to different types of networks, which raises the cost of connection, and reduces the efficiency of the utilisation of the network

In the 1970's, Integrated Service Digital Network (ISDN) was introduced to merge voice, data and image transmission onto one network Thus, the user would only need to make one connection and could subscribe to various telecommunications services using one line and one subscription number, increasing the efficiency of the network

But basic rate ISDN and primary rate ISDN networks will only support network speed lower than 2Mbps and can not provide the higher speed required for full motion picture transmission, LAN-LAN communication, and high speed data transfer Another restriction is that they only support circuit switched mode for speed higher than 64kbps and it will be very hard to accommodate new services in the future Hence, Broadband ISDN (B-ISDN) networks have been introduced [1]

B-ISDN networks are designed to support a variety of different transmission speeds, duration and burstiness They must provide

1 According to Minoru Akyiyama, at least 135Mbps interface speed, being capable of providing services of arbitrary transmission rate
2 The capability of providing various connection modes
3 Very low cell loss, cell delay, and very little distortion

4 The capability of providing Constant Bit Rate as well as Variable Bit Rate services

To incorporate all these features, ATM *Asynchronous Transfer Mode* is used in Broadband ISDN networks for signal transmission and switching

In parallel, the Internet is developing at a very high speed, with millions of individual users as well as universities, institutions and offices connected to it Even though the Internet is not commercialised for individual users, it is still possible that users of Internet services such as web browsing, file transfer, and Internet phone will be billed in the future when the Internet is completely privatised

Some figures will give people some impression of how this will be inevitable in the future The NSF (National Science Foundation) subsidy for NSFNET and various regional networks amounts to $20,000,000 per year, of which about $12,000,000 is for NSFNET The total cost of the Internet is estimated to be about $200,000,000 per annum (by the year 1995) [2] It is obvious from these figures that the maintenance cost of the Internet is huge, and, sooner or later, the governments that are funding the Internet now shall to recover the costs through charges and pricing schemes

From the analysis above, it is clear that investigation into ATM and Internet charging and pricing is essential for future networks The purpose of this thesis is to investigate charging schemes at both service and network level Two field trials are carried out to evaluate different charging schemes for this purpose

1.2 Objectives

The purpose for this project is to investigate charging and billing for the IP (Internet Protocol) over ATM both at the service level, and at the network level

The objectives are

1 To review all existing charging schemes for ATM networks and the Internet, and to evaluate them

2 To design a World Wide Web server system capable of storing charging and billing related information

3 To investigate service level charging schemes by applying different charging schemes to Internet application (World Wide Web browsing), and examine data obtained from a field trial

4 To investigate network level charging schemes by analysing traffic data obtained from a router

5 To examine the impact of different charging schemes

1.3 Summary

This thesis will start by briefly introducing the basics of ATM networks, HTTP (Hypertext Transfer Protocol) and IP, giving the reader a technical background of ATM, IP protocol and how they might work together This knowledge is necessary for the understanding of the principles of the work of this thesis and the foundation for the field trials The readers will have a basic knowledge of ATM networks and IP layers, when they come to the later chapters

In the third chapter, a review of current charging schemes is presented to the readers This review will incorporate various views which envisage both the Network Provider's (NP) and also the Service Provider's (SP) point of view Charging schemes will be introduced in these two categories Since most of the existing charging schemes are for network level charging, this category is given more emphasis in this chapter, with schemes for both ATM networks and the Internet listed While the emphasis of this thesis is to investigate charging for IP over ATM networks, current schemes that fall into this category are also listed But since it is a relatively new topic, and ATM is still at experimental stage, not many schemes can be found, and even with the listed ones, most of which are just recommendations The purpose of this chapter is to give the reader an impression of various charging schemes that are being used or recommended Some of these schemes are to be applied to data collected from a field trial, and will be further analysed

Investigation of network level charging is discussed in the fourth chapter Traffic of a small LAN is analysed and billing and charging schemes are examined in a small LAN scenario

The implementation of the WWW server as well as the system design for the charging and billing system will be introduced in the next chapter The reader will first be introduced to a wide range of WWW servers, from which one will be as the desired WWW server engine for the field trial

The server's internal structure will be analysed so that the reader will have a basic understanding of the program, how it works and how it reacts to a user request Then the necessary modifications made to the server source code will be shown, and functionality of the modifications explained Relevant information will also be given in succeeding sections

In the system design section, the infrastructure of the WWW charging and billing system will be analysed, and the readers will be shown how it conforms to ISO/IEC recommendations

The sixth chapter will be devoted to service level charging scheme examination This work is done through the examination of a field trial (Fionn) The field trial will be explained in detail, including its set-up, configuration, hardware and software environment, etc The results from the field trial will also be analysed, and conclusions drawn from these results

Chapter seven develops conclusions to the thesis

1.4 Judging Criteria

When choosing a suitable charging scheme, different criteria can be applied, but the best charging scheme should always be easily accepted by both network/service providers, as well as customers This implies that the charging scheme should be fair to NP/SP, and to the customers at the same time (offers the customers a low rate, as well as making a profit for the NP’s and SP’s) Hence, fairness is the most important feature that a charging scheme should have Any scheme that is fair to both sides could be accepted as a charging scheme by the NP/SP We’ve also chosen two other criteria when choosing our charging scheme

1 **Simplicity** Simplicity means little or no change to existing network system in order to use the scheme Thus systems using our charging schemes will no suffer from high implementation fee It also means that the scheme itself should be easy to understand, and to calculate
2 **Predictability** Predictability means that a user should be able to give a fairly close estimation of the charge before he/she receives the bill Thus network charges could always be kept at an acceptable value by cutting off unnecessary network usage
3 **Fairness** Apart from simplicity and predictability, a charging scheme should also be fair to all users of the network, i e people who uses the network a lot should not be paying significantly less than other users Only then, will a charging scheme be acceptable to customers

Chapter 2. General Background

The purpose of this chapter is to introduce the basics of ATM networks and IP protocol, including the structure of ATM cells and IP packet headers, since these control information other than user data play a subtle role when usage based charging schemes are applied HTTP protocol is also introduced here, as this is the protocol for WWW services It is important to understand the mechanisms of HTTP protocol and why the service level charging scheme had to be chosen as is

2 1 ATM Network

ATM stands for *Asynchronous Transfer Mode* It is primarily driven by telecommunications companies and is a proposed telecommunications standard for Broadband ISDN networks

2 1 1 STM Network

ATM is the complement of STM, which stands for *Synchronous Transfer Mode*, which is used by telecommunication backbone networks to transfer packetised voice and data across long distances

STM is a circuit switched networking mechanism, where a connection is established between two end points before data transfer commences, and torn down when the two end points are done Thus the end points allocate and reserve the connection bandwidth for the entire duration, even when they may not actually be transmitting data

The way data is transported across an STM network is to divide the bandwidth of the STM links into a fundamental unit of transmission called time-slots or buckets These buckets are organised into a train containing a fixed number of buckets and are labelled from 1 to N The train repeats periodically every T time period, with the buckets in the train always in the same position with the same label

A large number of trains and a large number of total buckets going empty most of the time creates a significant wastage of bandwidth and limits the number of connections that can be supported simultaneously Furthermore, the number of connections can never exceed the total number of buckets on all the different trains This is the reason why ATM is introduced

The telecommunications companies are investigating fibre optic cross-country and cross oceanic links with gigabit/sec speed And they would like to send signals in an integrated way, both real time traffic, such as voice and hi-resolution video which can tolerate some loss but not delay, as well as non-real time traffic such as computer data and file transfer, which may tolerate some delay but not loss Hence, ATM is conceived, proposed independently by BellCore and some giant telecommunications companies in Europe

The main idea here was to, instead of always identifying a connection by the bucket number, just carrying the connection identifiers along with the data in any bucket The size of the bucket is kept small so that if any bucket gets dropped due to congestion, not too much data gets lost, and in some cases can easily be recovered

2 1 2 What is ATM

In the late 80's, people carried out many experiments and established various switching models in order to find a proper transfer mechanism for Broadband ISDN networks In the States, it was called Fast Packet Switching, or FPS, and in Europe, it was called ATD, Asynchronous Time Division In 1988, CCITT named this technology ATM in its blue book, and recommended it for B-ISDN networks

The main characteristic about ATM is simplified network functionality ATM networks do not provide data link layer functionality, and leave functionality such as error control and flow control etc to terminals Figure 2 1 below shows the comparison between packet switching, frame relay and ATM networks

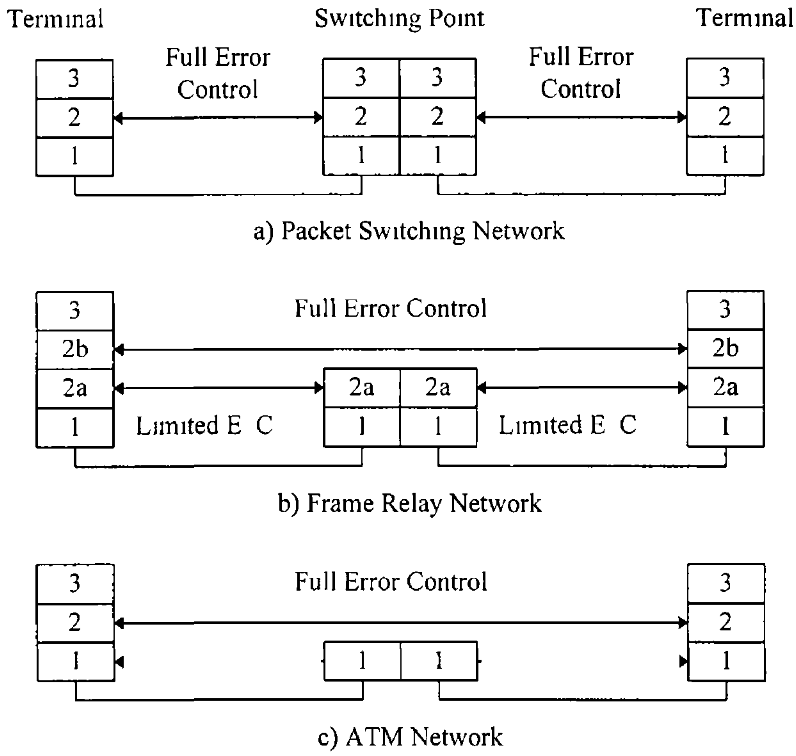


Fig 2 1 Functionality Comparison between packet switching, frame relay and ATM Networks

Fig 2 1 shows that packet switching network provides full functionality of the OSI layers 1-3 at the switching node, while frame relay provides only part of the layer 2 functionality, such as CRC check

2 1 2 1 ATM Mechanism

ATM is a transfer mode in which information is organised in the form of cells A cell containing certain pieces of information does not need to appear periodically, and it is in this sense that the transfer mode is asynchronous

ATM cells are of fixed length Each cell contains 53 octets The first 5 octets are called the header, and the 48 octets contain user information called the payload A more detailed description of cell structure is given in the next section

ATM networks use *asynchronous time division multiplexing* (shown in Fig 2 2)

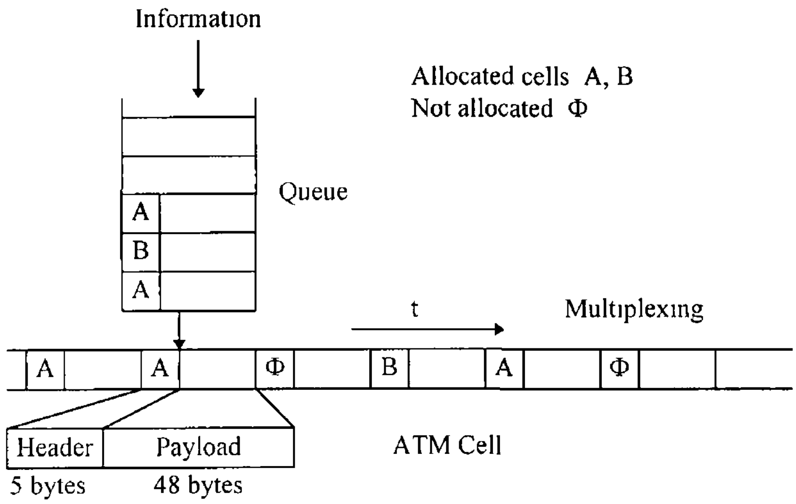


Fig 2 2 ATM Cell and Multiplexing

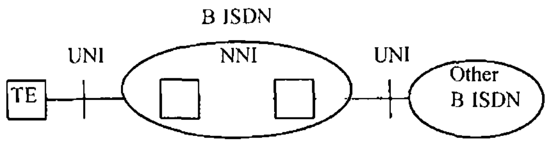
Cells from different information sources (different services and different senders) come together and queue in a buffer Cells in the buffer are sent to the transmission line one by one The header containing flags such as A and B represents the cell's destination

Because of the random nature of the information generated, cells arrive to the queue randomly as well Services with very high speed have many cells arriving in a very short time interval, while services with lower speed will have fewer cells arriving in the same time interval Cells containing the same flag (same destination) do not correspond to a fixed time slot, and do not have to appear periodically This provides ATM networks with great flexibility

2 1 2 2 ATM Cell Structure

In ATM networks, the data packets are called cells, and they consist of a cell header and an information field The header stores control information, which takes 5 bytes and the information field takes the remaining 48 bytes

The following diagram (Fig 2 3) shows the header structure at the User-Network Interface (UNI) and the Network-Network Interface (NNI)



bits

1	2	3	4	5	6	7	8	
GFC				VPI				1 b
VPI				VCI				2 y
VCI								3 t
VCI				PT		RES	CLP	4 e
HEC								5 s

UNI Header

bits

1	2	3	4	5	6	7	8	
VPI								1 b
VPI				VCI				2 y
VCI								3 t
VCI				PT		RES	CLP	4 e
HEC								5 s

NNI Header

Fig 2 3 ATM Cell Header Structure

GFC Generic Flow Control In order to control the flow of traffic on ATM Connections from a terminal to the network, a GFC mechanism is proposed by CCITT at the UNI This functionality is supported by GFC field in the ATM cell header

VPI, VCI Virtual Path Identifier, Virtual Channel Identifier ATM is a connection oriented transfer mode The header values are assigned during the connection set up phase and translated when switched from one section to another In routing, there are two types of connections, i e a virtual channel connection (VCC) and a virtual path connection (VPC) A VPC is an aggregate of VCCs Switching on cells is first done on the VPC, and then on the VCC VPI is analogous to LCGN in X 25, while VCI is similar to LCN in packet switching

PT Payload Type To identify whether the information field contains user information or network information 00 stands for user information, while other values are to be defined

CLP Cell Loss Priority When congestion occurs in a network, the cells that have CLP equal to 1 can be dropped Cells with CLP equal to 0 stand for high priority and can never be dropped by the network This is directly related to quality of service (QoS)

HEC Header Error Control Detects and prevents errors that may occur in the header due to disturbances

RES Reserved Field

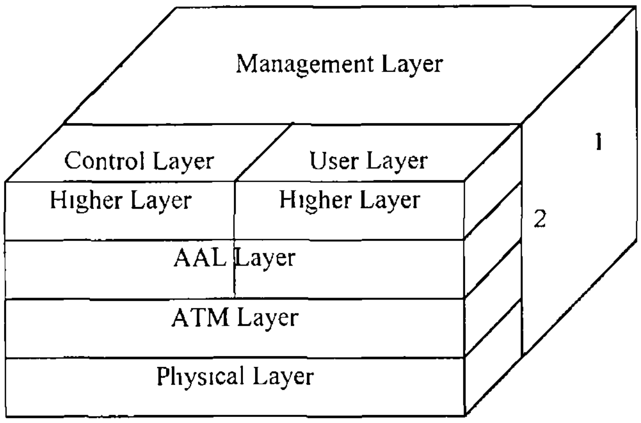
2 1 2 3 ATM Network Layers

ATM networks divide ISO OSI reference model's first layer (Physical Layer) into physical layer, ATM layer, and ATM Adaptation Layer (AAL) These layers can be further subdivided into sub-layers, e g AAL can be divided to CS (Convergence) and SAR (Segmentation And Re-assembly) sub-layers, physical layer can be divided into TC (Transmission Convergence) and PM (Physical Medium) sub-layers

The function of the AAL layer is to realise voice, graph, data and all kinds of different communications services Different services require different AAL layers

The CS layer takes service data units (SDUs) from the AAL's service access point (SAP)

The main function of the SAR layer is to re-segment and group data generated by the CS layer, to cut them into the required ATM cell length The length of data generated by SAR is 48 octets, with 46 or 47 octets effective payload



1 Plane Management
2 Layer Management

Fig 2 2 B-ISDN Reference Model

2.2 Internet Protocol

The Internet Protocol is the only one protocol associated with the complete protocol suite (stack) used with the Internet The complete suite, known as TCP/IP, includes transport and application protocols which are now used as the basis of many other commercial and research networks [9] The IP protocol that we refer to here in this section is IPv4

2 2 1 Internet Protocol

Internet Protocol (IP) is an Internet-wide protocol that enables two transport protocol entities resident in different end systems/hosts to exchange message units (NSDUs) in a transparent way This means that the presence of multiple, possibly different, networks or subnets and intermediate systems/gateways is completely transparent to both communicating transport entities As the IP is a connectionless protocol, message units are transferred using an unacknowledged best-try approach [9]

The IP protocol provides a number of core functions and associated procedures to carry out the various harmonising functions that are necessary when inter-working access dissimilar networks These include

- Fragmentation and re-assembly this is concerned with issues relating to the transfer of user messages across networks/subnets, which support smaller packet sizes than the user data
- Routing to perform the routing function, the IP in each source host must know of the location of the Internet gateway or local router that is attached to the same network or subnet Also, the IP in each gateway must know the route to be followed to reach other networks or subnets
- Error reporting when routing or re-assembling datagrams within a host or gateway, the IP may discard some datagrams This function is concerned with reporting such occurrences back to the IP in the source host and with a number of other reporting functions

2 2 1 1 IP Address Structure

There are two network addresses associated with a host/end system attached to an Internet In ISO terminology, these are the network service access point (NSAP) address and the subnet point of attachment (SNPA) address With TCP/IP, these are the IP address and the network point of attachment subnet type, whereas the IP address is a unique Internet-wide identifier The structure of an IP address is shown in Fig 2 3 below

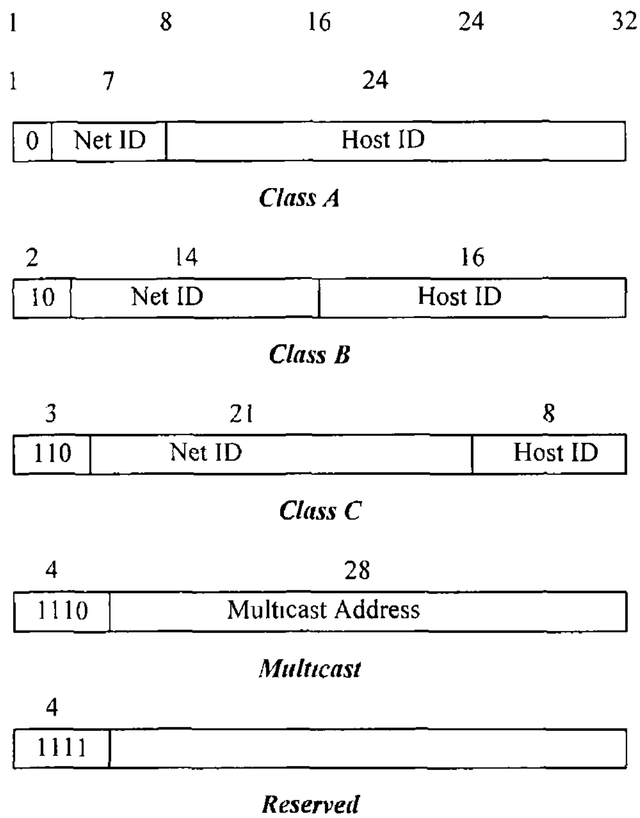


Fig 2 3 IP Address Formats

Each IP address is a 32-bit integer to ensure that all hosts have a unique identifier Three different address formats are then defined to allow for the different sizes of network to which a host may be attached Each format is known by an address class A single Internet may use addresses from all classes The three primary classes are A, B and C Each is intended for use with a different size of network The class to which an address belongs can be determined from the position of the first zero bit in the first four bits The remaining bits specify two fields a network identifier and a host identifier The field boundaries are located on byte boundaries to simplify decoding For example, the machine *odin* has IP address 136 206 36 10, and Dublin City University's network is on subnet 136 206 0 0

2 2 1 2 IP Protocol Data Unit

The IP data unit is know as a datagram The format and contents of a datagram is shown in Fig 2 4 below

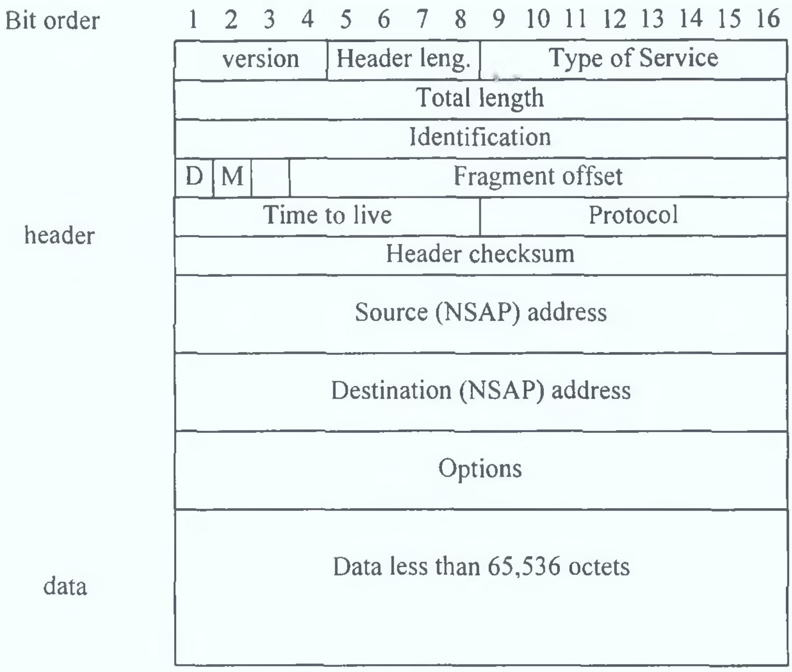


Fig. 2.4 Internet datagram format

We can see from the above diagram that the IP header field takes 20 octets excluding the option field, which is quite lengthy. Attention should be paid to this lengthy header, as the IP header plays an interesting role in charging as we will explain in detail in chapter 6.

The IP header provides source and destination IP addresses for each packet, and the position of each packet in relation to the initial user data message, which is important for the re-assembly procedure.

2.2.2 IP over ATM

The Internet is growing everyday at an astonishing speed, and while more and more users join the Internet everyday, network traffic is becoming heavier and heavier. Since ATM technology offers its users a very high speed, it is very natural that people would like to use ATM technology in the Internet. In fact, the deployment of ATM into the Internet community is just beginning and will take many years to complete.[10]

As we have discussed in the previous sections, ATM is designed for switching short fixed length packets in hardware over Gigabit/sec links across very long distances. Thus its place in the protocol stack concept is somewhere around the data link layer. However it does not cleanly fit in to the abstract ISO OSI layered model, because within the ATM network itself, end-to-end connection, flow control, and routing are all done at the ATM cell level So there are a few aspects of traditional higher layer functions present in it In the OSI reference model, it would be considered layer two (where layer one is the physical layer and layer two is the data link layer in the Internet protocol stack) But it is not very important to assign a clean layer name to ATM, so long as it is recognised that it is a hardware implemented packet switched protocol using 53 byte fixed length packets

What is perhaps more relevant is how will all this interact with current TCP/IP and IP networks in general, and with applications which want to talk to ATM directly in particular A convenient model for an ATM interface is to consider it another communications port in the system Thus from a system software point of view, it can be treated like any other data link layer port Thus for instance, in IP networks connected via gateways to ATM backbones, the model would be of no difference than it presently is for a virtual circuit connection carried over an STM link The exception is that an IP packet over an ATM network would be fragmented into cells at the transmitting UNI, and reassembled into the IP packet at the destination UNI Thus a typical protocol stack might look like this

Data
TCP
IP
ATM Adaptation Layer
ATM Data Link Layer
Physical Layer (SONET)

Fig 2 5 TCP/IP over ATM

Thus, just like an Ethernet port on a host is assigned an IP address, the ATM port may also be assigned an IP address The IP software in a router decides which port to send a packet to based on the IP address, and hands the packet to the port The port then does the right thing with it For an Ethernet port, the Ethernet header is tacked on and the frame transmitted in Ethernet style Similarly, for an ATM port, the IP datagram is fragmented into cells for which an ATM adaptation layer is specified in the standards The fragmentation and re-assembly is done in hardware on the sending and receiving sides A VCI label acquired via an initial one-time connection establishment phase is placed in the header of each cell, and the cells are drained down the ATM data link layer pipe On the receiving side, the cells are reassembled in hardware using the ATM adaptation layer, and the original IP packet is reformulated and handed to the receiving host on the UNI The adaptation layer is not a separate header, but is actually carried in the payload section of the ATM cell [11]

2.3 Hypertext Transfer Protocol

The Hypertext Transfer Protocol (HTTP) is an application level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands) A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred [8]

The reason for introducing HTTP here is because it has been used by the World Wide Web global information initiative since 1990 The version of HTTP that is introduced here is HTTP/1 0

The HTTP protocol is based on a request/response paradigm A requesting program (termed a client) establishes a connection with a receiving program (termed a server) and sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content The server responds with a status line, including its protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta-information, and possible body content It should be noted that a given program may be capable of being both a client and a server The use of those terms refers only to the role being performed by the program during a particular connection, rather than to the program's purpose in general

On the Internet, the communication generally takes place over a TCP/IP connection The default port is TCP 80, but other ports can be used This does not preclude the HTTP/1 0 protocol from being implemented on top of any other protocol on the Internet, or on other networks HTTP only presumes a reliable transport, any protocol that provides such guarantees can be used, and the mapping of the HTTP/1 0 request and response structures onto the transport data units of the protocol in question is not specified in the draft

The HTTP protocol requires that the connection be established by the client prior to each request and closed by the server after sending the response Both clients and servers must be capable of handling cases where either party closes the connection prematurely, due to user action, automated time-out, or program failure In any case, the closing of the connection by either or both parties always terminates the current request, regardless of its status

Chapter 3. Review of Charging Schemes

A charging scheme can be expressed in the form:

$$y=f(p_1, p_2, \dots, p_n)$$

Where $p_1,p_2,\dots p_n$ are charging parameters, and y is the charge produced by this particular scheme. The value of n depends on the choice of scheme, and parameters vary from scheme to scheme. Most common charging parameters include: average bandwidth, effective bandwidth, peak bandwidth, duration of call and cell loss ratio etc.

Various charging schemes, both network level and service level for ATM networks and/or IP network are discussed in this chapter. The purpose is to compare all the mechanisms, list the advantages and disadvantages of each method. One of the schemes is chosen as the network level charging scheme in our field trial using the criteria described in introduction.

3.1 Existing ATM Network Level Charging Mechanisms

As ATM networks are not commercialised, all the published charging schemes for ATM networks are just proposals. Most of these charging schemes, especially effective bandwidth ones require many network traffic parameters and some of them are very difficult to measure using existing technology.

3.1.1 Charging Scheme Based on Statistical Measurements

Description

Statistical measurements are measurements involving statistical figure of network, e.g. average bandwidth in a certain time period, cell loss ratio in certain period, etc. The algorithm discussed here is presented by Vierø [15]. Every h seconds (*scanning* or *sampling* interval), the traffic offered by each connection is sampled for a *duration* of T_S seconds ($T_S \leq h$). For this period of T_S seconds, the number of cells transmitted is monitored (each call measured individually), thus yielding a value for the average bandwidth during this period (See Figure 3 1) (Individual connections may require different scanning times and thus may need their own cell counters)

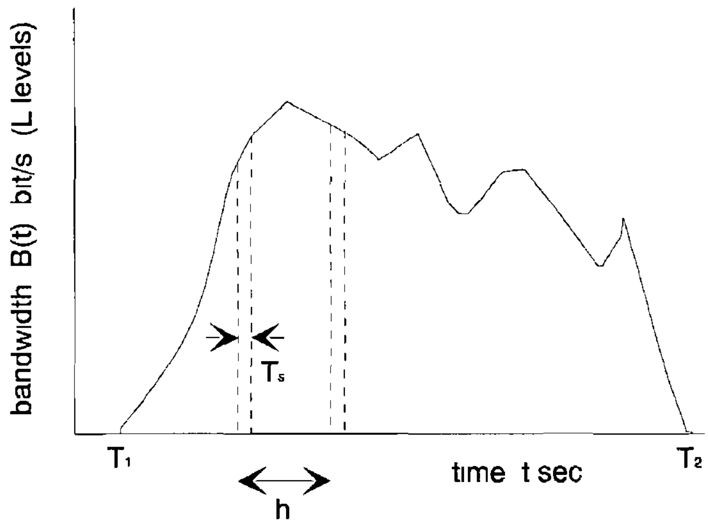


Fig 3 1 Parameters for Statistical Charging

Choice of Sampling Interval and Duration

To try and ensure that the method does not miss traffic burst, h and T_S should be chosen carefully as follows

h The sampling period is $h = \pi/\Omega$ seconds, where Ω is the angular frequency in radians per second This must be chosen to take account of the fact that according to sampling theory, all information outside the frequency band $[-\Omega, \Omega]$ is lost

T_S The minimum time we can have for the sampling duration T_S is the time T_C it takes for one cell to pass the measurement mechanism If the subscriber loop has the access rate C and the cell length is b_C, we have $T_C = b_C/C$ If $h = T_S = T_C$, we would only be able to measure the rate 0 or C, such a choice would make it impossible to detect individual traffic bursts (since during this time period T_C, either 0 cell or 1 cell can pass the sampler) Hence we want T_S to be small enough so that we can detect a suitably small part of a burst We can therefore let $T_S = \beta b_C/C$, with $\beta = 20$ as a possible value for the constant

Statistical Description

To facilitate subsequent computation, we may consider the maximum possible bandwidth for a given connection to be quantised into one of L levels, where $L \geq 2$ For N samples, the total connection time is in the range Nh to (N+1)h seconds, the mean bandwidth is given by 3 1 [15]

$$\mu = N\sum_{j=1}^{L} jp_j \qquad 3\ 1$$

Statistical Justification

It can be shown that the mean bandwidth defined above also equals to the expectation of the bandwidth

If the values of j occur with the probabilities $p_0, p_1, \ldots, p_L$, where $\sum_{j=0}^{L} p_j = 1$,

We can assume that for N samples, the distribution will be (L+1)-normal Thus the expectation of the charge in terms of charging units is [15]

$$\mu = E[0x_0 + 1x_1 + \ldots + Lx_L] = N\sum_{j=0}^{L} jp_j \qquad 3\ 2$$

Alternative Expression for the Mean Bandwidth or Charge

We can express the charge μ in terms of

(1) The system time available to the user, namely the terms outside the summation (3 3), where h(L/C) is the fraction of system time per sample and N as defined before, and,

(2) The mean bandwidth used, given by the summation of the product of each discrete bandwidths (1, ,L) and the probability of its occurrence

$$\mu = Nh\frac{L}{C}\sum_{j=1}^{I} jp_j \qquad 3\ 3$$

More General Charging Function

Every h seconds, the following more general charging function may be invoked

$$c = k_1 * J + k_2, \qquad 3\ 4$$

where,

(a) J is the average bandwidth, measured from number of cells counted over a duration of T_S seconds, and.

(b) k_1 and k_2 are integer constants.

Possible choices are $k_1 = 1$ and $k_2 = 0$, although it might be desirable for administrations to have $k_2 > 0$, in order to give the customer an incentive to close down the connection, if it is not actually used over a longer period of time.

MEASUREMENT:

- Only need to measure the bandwidth.

ADVANTAGE:

- Easy to implement

DISADVANTAGES:

- This method, while obtaining a measure of cells passed, does not take account of any contract between user and provider.
- The algorithm could miss a burst of traffic. Malicious users could take advantage of this.

COMMENTS:

- This method assumes that the total number of cells passed may be assessed adequately by only counting a fraction of those cells actually passed. This may not always be valid.
- Even though we restricted $L \geq 2$, but the derivation above still applies when $L=1$. In this case, this method will become a generalisation of the well-known Karlsson Scanning Principle; however, the latter approach can use random scanning intervals, which can influence the accuracy of the traffic measurements.

3.1.2 Effective Bandwidth Charging Schemes

Introduction

The effective bandwidth of variable bit rate source is a natural basis for charging as it summarises all statistical characteristics of the source into only one parameter. After this such source can be considered as a constant bit rate source with the mean rate equal to the effective bandwidth So it could be charged as constant bit rate source

There are more complicated charging schemes based on effective bandwidth In the following, the charging scheme proposed by F P Kelly [12] which is also based on effective bandwidth will be reviewed The subsequent section considers a practical method of estimating effective bandwidth by measurement due to C Courcoubetis et al [14] Some performance results on charging schemes based on effective bandwidth are presented The implementation of the usage based charging for real time VBR is discussed

3 1 2 1 Effective Bandwidth Scheme

This section considers two papers by F P Kelly, namely “Tariffs, Policing and Admission Control for Multiservice Networks” [12] and “Charging and Accounting for Bursty Connections” [13] Our review concentrates on the first paper but adds some extra comments on the second paper, which is similar

Description

How can a network encourage the co-operative sharing of network resources by users using tariff structures? The underlying idea proposed in the paper is that at call admission, a contract would be made between user and network specifying in more or less detail the statistical properties of the call, and the policing mechanisms would endorse the contract

In the paper the problem of finding such tariff structures is addressed For example, in the case of on/off sources with policed peak rate, the tariff structure takes the form a charge a_m per unit time and a charge b_m per cell carried The pair (a_m, b_m) are fixed by a declaration of the expected mean rate of the source m made by the user at the time of call discussion But in general case of a bursty source, the tariff structure is more complex In particular, the tariff structure does not operate in well-known terms for users such as the mean bit rate

The idea of introducing such tariffs can be illustrated in the case of on/off sources with policed peak rate as follows This simple model is chosen only for illustration purposes The model can be generalised on more realistic cases

We consider a very simple on/off source X with peak rate h and mean rate m At each moment of time the source produces h cells with probability m/h and with probability $1-m/h$ it does not produce any cells,

$$P\{X=0\}=1-m/h, \quad P\{X=h\}=m/h \qquad 3\ 5$$

We shall assume that multiplexer has no buffer and serves with rate c per unit time Suppose we want to know n, the maximum number of on/off sources $X_1, \ldots, X_n$ with the same distribution as X which can be served by the multiplexer in order to satisfy the quality of service q

$$P\{ X_1 + \ldots + X_n > c \} < q \qquad 3\ 6$$

Then, it is known that in this case the number of sources can be found using $C/B(X)$, where $B(X)$ is called the effective bandwidth of the source and is equal to [12]

$$B(X) = \frac{1}{\alpha}\log E e^{\alpha X} = \frac{1}{\alpha}\log\left[\frac{m}{h}e^{\alpha h} + \left(1-\frac{m}{h}\right)e^{\alpha 0}\right] = \frac{1}{\alpha}\log\left[1+\frac{m}{h}\left(e^{\alpha h}-1\right)\right], \qquad 3\ 7$$

for some parameter α, which is defined by the distribution of X and parameters c and q, E denotes the expectation with respect to the distribution of random variable X Note that parameter α depends strongly on the distribution of random variable X and parameters c and q

The derivation of formula (3 7) for un-buffered model can be found in [17] (formula (2 6)) or in [18] for some other models The formula is based on the so-called *large deviation approximation* Note that we do not specify precisely the base of the logarithm in log( ), so the constant α will also depend on the choice of the base For example, we can assume that log=ln in what follows

Next, we consider a little more complex on/off source. Suppose that for the user the peak rate h cells per unit time is known and fixed. But its mean rate M is not known with certainty. But suppose a user who is about to make a call, has a prior distribution G for the mean rate M of a call. The user might, for example estimate the distribution G by recording the observed mean rates on past calls. Then the expected mean rate of the call is $E_G\, M$, where E_G denotes the expectation with respect to the distribution G. The effective bandwidth of this source is equal to

$$B(X) = \frac{1}{\alpha}\log E e^{\alpha X} = \frac{1}{\alpha}\log\left[1 + \frac{E_G M}{h}\left(e^{\alpha h} - 1\right)\right], \qquad 3\ 8$$

We suppose that the user and network interact in the following way. Before a call admission, the user announces a value m, and then charges for the call an amount $f(m,M)$ per unit time, where M is the measured mean rate for call. The tariff function $f(,)$ is known for the user and the network. Then, because the mean rate M is not known but its distribution G is known, the user in order to minimise the cost of call might attempt to select m so as to minimise the expected cost $E_G\, f(m, M)$ per unit time. Let **m** be an optimal declaration of m for the user. What properties would the network like the optimal declaration **m** have?

- The network would like to be able to deduce from **m** the user expected mean rate $E_G\, M$, i e **m** $=E_G\, M$, and hence the effective bandwidth (3 8) of the call
- The network would like that the expected cost per unit time under the optimal declaration **m** be proportional to the effective bandwidth of the call (or, equivalently, equal to the effective bandwidth under a choice of units, i e E_G **f**$(m,M)=B_G\,(X)$)

A remarkable property of these two requirements is that the tariff **f**(m, M) has to be defined as

$$\mathbf{f}(m, M) = a_m + b_m M, \qquad 3\ 9$$

where parameters a_m and b_m are uniquely determined by the condition that (3 9) is the tangent of the curve

$$B(M) = \frac{1}{\alpha}\log\left[1 + \frac{M}{h}\left(e^{\alpha h} - 1\right)\right], \qquad 3\ 10$$

at the point $M=m$ [12]

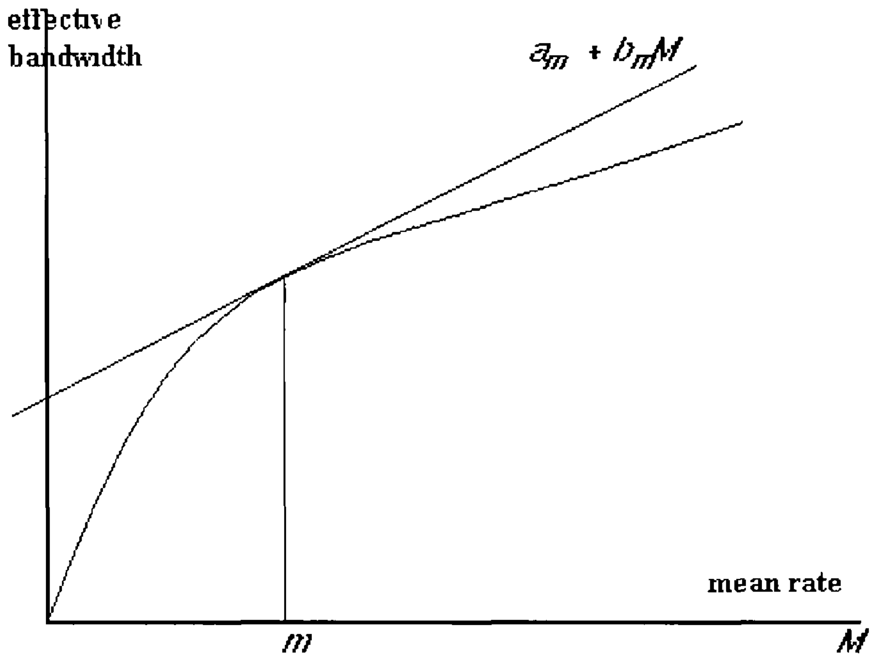


Fig 3 3 Effective bandwidths for varying mean rates, together with the linear tariff selected by the declaration **m.**

If it is not required that the expected cost per unit time under the optimal declaration is equal to the effective bandwidth, but only require that the optimal declaration for the user be the mean **m**=E_G M, then many tariff structures are possible For example, for any differentiable concave function $F(M)$, let

$$\mathbf{f}(m, M) = a'_m + b'_m M \qquad 3\ 11$$

be the tangent to the curve $F(M)$, at the point $M{=}m$ Then the optimal strategy for the user under the tariff structure **f** is to declare the mean **m**=E_G M

All these result essentially used the independence assumption for on/off sources This is not the case for ATM sources because of the burstiness The independence of the loads produced by different sources is an essential constraint of the above analysis also

The simplified model can be easily generalised on sources with unknown peak rate h and mean rate m (see Section 3 of [12]) Moreover, formally it can be generalised on some bursty sources (see Section 5 of [13])

In [13] the author proposed a mechanism to vary the declaration of a parameter Changes in the tariff regime might be made over time interval than round trip delay time, but shorter than a connection This mechanism can be described as follows Suppose that a connection may change its choice of tariff regime at any time, but must pay a fixed charge *d* for the set-up costs incurred by the network for each connection Then a user is able to make its own evaluation of the benefit of changing the choice of tariff

Measurements

In order to implement the method, the network has to estimate statistically a few points of an "effective bandwidth curve" for each source

Effectiveness

Advantages

- It is usage based charging
- The method of tariffing is flexible because the user can correct the declaration of some parameters during calling,

Disadvantages

- The Network can not use parameters declared by the user for finding of effective bandwidth because he could make an error in the declaration
- Generally, the network has an advantage over customers, because the customers have to pay some extra money for errors in declaring parameters This can create an "unfriendly environment" for users
- In order to implement the method network has to statistically estimate an "effective bandwidth curve" of each source If a source is time homogeneous and stationary or close to stationary, the estimation can be done quickly and efficiently For non-stationary sources to estimate the effective bandwidth is much more difficult

3 1 2 2 Effective Bandwidth Estimation

Description

This review is based on the paper C Courcoubetis, G Kesidis, A Ridder, J Walrand and R Weber "Admission Control and Routing in ATM Networks using Inferences from Measured Buffer Occupancy " [14]

In the paper the issue of call acceptance and routing in ATM networks is addressed The main goal is to design an algorithm that guarantees bounds on the fraction of cells lost by a call because of buffer overflows An original method for call acceptance and routing in ATM is proposed The important feature of the method is that it does not require models describing the statistical characteristics of the traffic The main idea of the method is based on the estimation of effective bandwidth using scaling properties of the cell loss probability and parallel simulations of virtual switches

In order to guarantee that new additional calls will not violate the cell loss ratio, the switch has to estimate its spare capacity The difficulty to do this is that the switch has to evaluate the value of the cell loss ratio would have if more calls were using that switch This could be done by simulations, but to implement such simulations is not so simple we have to use records of previous calls and use them for simulations Moreover, there is another difficulty since the loss probability is very small, estimators based on fractions of lost cells have a very large variance and are very slow To reduce the variance and organise simulations in parallel (and in more efficient way), we estimate the value that the cell loss probability would have if the switch buffers could store fewer cells That is, we keep track of a virtual buffer occupancy corresponding to a smaller buffer capacity To relate the statistics of cell losses of this smaller buffer to those of the actual buffer we use results on the shape of the loss probability function of the buffer size and its scaling properties

Monitor to Inter Network Overflow Statistics (MINOS)

We describe an algorithm that can be used by a switch to predict its spare capacity This algorithm is a Monitor to Inter Network Overflow Statistics (MINOS) The algorithm consists of a few steps

Step 1 We want to estimate the loss probability in the switch buffers Consider a given buffer of size B cells, with N virtual circuits sharing the buffer, and served by a fibre with transmission rate c cells per second Let $F(N, B, c)$ be the fraction of cells lost due to buffer overflows We want to estimate the fraction $F(N(1+\varepsilon), B, c)$ of cells lost when εN more calls are added It is proved that the cell loss probability F has the following (scaling) property for large B

$$F(N(1+\varepsilon), B, c) \approx F(N, B, c/(1+\varepsilon)) \qquad 3\ 12$$

We note that (3 12) means that the fraction

$$\frac{\log F(N(1+\varepsilon), B, c)}{\log F\big(N, B, c/(1+\varepsilon)\big)} \qquad 3\ 13$$

is approximately equal to 1 [12]

Thus, we can estimate F, when a fraction ε more calls are added by estimating F with the current number of calls, N, and the service rate reduced by the same fraction Why to estimate $F(N, B, c/(1+\varepsilon))$ is simpler than $F(N(1+\varepsilon), B, c)$? The main reason is that this estimation can be done in parallel with normal work of the switch

How this can be done? To estimate $F(N(1+\varepsilon), B, c)$, a device is added to the switch that keeps track of the buffer occupancy, $X(t)$ at time t, when the service rate is $c/(1+\varepsilon)$) Specifically, when a cell arrives at the buffer, $X(t)$ is incremented by one Also, $X(t)$ is decremented by one every $(1+\varepsilon)/c$ time unit when $X(t)>0$

Step 2 But the problem of the *reliable* estimation of $F(N, B, c/(1+\varepsilon))$ is more complex This value is very small (typically about 10^{6}-10^{9}) Consequently, a direct estimator based on the fraction of lost cells has a very large variance To improve the estimator, the device will estimate the losses for smaller buffers (called virtual buffers) so as to increase the frequency of buffer overflows, and then speed up the collection of sufficient number of "important" samples There is a trade-off in choosing the size of the virtual buffers

- If these virtual buffers are still too large, our estimates will be too slow

- However, if these virtual buffers are too small, the original system is over-distorted and we have a large error when we extrapolate back to original buffer size B

The reasonable compromise is to chose the sizes of virtual buffers of the same order as B More specifically, let B/k be the size of a virtual buffer for some $k>1$ From the theory of large deviation it is known that

$$F\left(N, B, \frac{c}{1+\varepsilon}\right) = \exp\left(-B \times I(N, \frac{c}{1+\varepsilon}) + o(B)\right) \qquad 3\ 14$$

where $I(x, y)$ is some positive concave function In our method we do not need to know this function for all values x, y

We shall assume that the factor $e^{o(B)}$ to be the form $A \times B^{\xi}$ for some constants $A>0$ and ξ Then we obtain

$$F\left(N, \frac{B}{k}, \frac{c}{1+\varepsilon}\right) = A \times \left(\frac{B}{k}\right)^{\xi} \times \exp\left(-B \times I(N, \frac{c}{1+\varepsilon})\right) \qquad 3\ 15$$

As parameters N, c, ε are fixed, then we have three unknown variables A, ξ and $I(N, c/(1+\varepsilon))$ We will carry out this estimate for three values of k $k_1>k_2>k_3>1$ These three non-linear equations can be solved for variables A, ξ and $I(N, c/(1+\varepsilon))$ We can then plug in variables A, ξ and $I(N, c/(1+\varepsilon))$ into (3 15), the expression for $F(N, B, c/(1+\varepsilon))$, and thus compute the desired quantity $F(N(1+\varepsilon), B, c)$

To summarise the above, the estimation algorithm in the device keeps track of three "virtual buffers" occupancy processes with buffers of size B/k_i, where $i=1, 2, 3$, with service rate $c/(1+\varepsilon)$) and the number of calls N These computations can be done in parallel with the normal operation of the switch

Step 3 Routing algorithm The method of estimation of the cell lost probabilities can be used for the routing algorithm For simplicity, we assume that all the virtual circuits carry calls of the same type Denote $F_n = F_n(N_n, B_n, c_n)$ the current fraction of cells lost at buffer n, for all buffers n in the network Assuming a first-come-first-serve queuing discipline in each buffer, F_n is the fraction of cells lost at buffer n by each call that uses buffer If call i uses buffers 1,2, ,m, the fraction of cells lost by that call is

$$1-\prod_{n=1}^{m}(1-F_n) \approx \sum_{n=1}^{m} F_n \qquad 3\ 16$$

Suppose we are trying to route a new call Using the above method, buffer n estimates $F_n' = F(N_n(1+\varepsilon), B_n, c_n)$ We attempt to find a path for the new call that satisfies

$$G^{new} \geq \sum_{n \in path} F_n', \qquad 3\ 17$$

where G^{new} is the fraction of lost cells acceptable to the new call Moreover, the router must ensure that, by choosing a particular path for the new call, the above constraint is not violated for any existing, previously routed call i (with guarantee G^i), which uses all or part of that path If no path is found that satisfies these constraints, the new call is refused

Measurements

In order to use the method, the network has to implement for each switch a device for simulations in parallel with functioning of the switch This device can, for example, consist of an additional microcomputer with simple controlling software Such a device is able to shed light on any hardware-related implementation problems that may arise with the switch

Effectiveness

Advantages

- The method does not require models describing the statistics of the traffic
- The method of call acceptance described in this paper can be used with various switch architectures (e g output-buffer, shared-buffer, Batcher-Banyan)

Disadvantages

- Some results are only heuristic In particular, the representation (3 15) is not proved theoretically, but it is a very natural hypothesis,
- The solution of three non-linear equations for different buffer occupancies can also give a large deviance This problem is not discussed in the paper

Comments

It is an interesting proposal. The method can be implemented for the estimation of effective bandwidth and, in particular, for estimation of "effective bandwidth curve". So it can be also useful for charging based on effective bandwidth.

3.1.3 Other Charging Schemes

3.1.3.1 Three Term Charging

The equation for this scheme is as follows.

$$C = D + E \times t + F \times v \qquad 3.18$$

Where the charge C is composed of:

1. a constant term D consisting of a subscription charge and a short term access charge,
2. a duration dependent term (E.t), where E is the charge per unit time and t is the duration or lifetime of the connection, and,
3. a volume dependent term (F.v) where F is the charge per cell, and v is the total number of cells passed by the connection.

While the constants D, E and F are likely to be contractual, the scheme contains no contractual obligation regarding either the total number of cells to be passed by the connection or the maximum rate at which these may be passed. Hence the charging scheme, as it stands, can offer the customer surprises regarding the resultant total charge for the connection.

Now if the cell rate is limited, then the maximum total volume is linearly related to time, and the terms E.t and F.v can be replaced by a term G.t, where G is a constant. If the cell rate limitation is a hard boundary, the scheme is then equivalent to the duration-based charging approach normally adopted by the POTS network operators. If the cell rate lımıtatıon ıs a statıstıcal (e g Poıssonıan) boundary, the scheme ıs equıvalent to Envelope Chargıng In both cases we then have a scheme whıch offers "no surprıses" to the customer, who merely needs to keep track of tıme ın order to calculate the charge levıed on hım Essentıally the customer was payıng a fixed charge per unıt tıme that the gıven user ıs actually usıng the network The network operator ıs also pleased because he can predıct the worst case effect the connectıon may have on the traffic condıtıons ın the network, knowıng that the user can not exceed the bandwıdth lımıt (hard or statıstıcal) and may, ındeed, use less than the maxımum bandwıdth allotted to hım

A scheme contaınıng a volume dependent term but no tıme dependent term would suıt users of avaılable bıt rate (ABR) category servıces e g those transferrıng files or lıbrary ınformatıon overnıght, where access and transfer tımes are not ımportant but space prıorıty (no cell loss) ıs very ımportant

An advantage of the three-term scheme ıs that ıt ıs sımple enough to be ımplemented easıly

3 1 3 2 Resource Prıcıng - Dynamıc Chargıng

Descrıptıon

Thıs revıew ıs based on the paper S Low and P Varaıya, "An algorıthm for optımal servıce provısıonıng usıng resource prıcıng" [16]

The authors descrıbe a method of usıng resource prıcıng to ensure that servıces are provıded to customers ın an ATM network ın an optımal way The two types of resources consıdered are bandwıdth and buffers The authors consıder these quantıtıes to be "substıtutable resources", whıch means that a customer may be happy wıth the servıce provıded wıth dıfferent combınatıons of bandwıdth and buffers The balance of these two types of ıesource can thus be varıed to optımıse the provısıon of servıces to customers

The authors assume that a finıte set of servıces ıs avaılable on the network They appear to define a servıce ın terms of a mean holdıng tıme, a mınımum bandwıdth requırement, and a buffer requirement (which is a function of the bandwidth) The aggregate demand of users for a given service is assumed to decrease exponentially with price

The minimum value μ_{min} of bandwidth which a customer may request is that even with infinite buffers available, ensures zero loss, but results in an unacceptable delay when the service is used There is also a maximum value, μ_{max} that, if exceeded, brings no benefit to customers, because their data never require buffering

The customer is required to know these two quantities, and also the function $b(\mu)$, the amount of buffers required to ensure that, if the connection bandwidth is μ, the user's data is transmitted without loss (or with an acceptable level of loss)

The customer is presented with a rent by the network operator of α/unit for bandwidth and β/unit for buffers It is not clear from the paper what time scale the rent would apply for The user accordingly rents resources in such a way as to minimise the cost of his service

- The values of rents are updated periodically (period unspecified) by the network operator The algorithm used is based on the following data for each service
- The demand for the service at a zero price
- The amount of bandwidth and buffers purchased in the previous period

The algorithm also requires knowledge of the buffer and bandwidth capacity associated with each network link

The optimality criterion used is the maximisation of the *social welfare,* defined as the sum of the *user surplus* and the *expected revenue* The *user surplus* is the sum (over all services) of the areas under the demand curve above the service price

Measurements

The algorithm requires monitoring of the reserved amounts of bandwidth and buffers for each service, and knowledge of the buffer space and bandwidth capacities of each link It is claimed that the algorithm obviates the necessity for traffic policing

Effectiveness

The algorithm bears strong similarities to the leaky bucket method of traffic policing, which uses the same parameters (leak rate = bandwidth, bucket size = buffer spaces) The issue of time scales is critical to the effectiveness of the method and requires further exploration If the update rate for prices is very short, the overhead may be prohibitive If it is long, there may be problems with bursty traffic sources Another implementation difficulty may relate to the number of services supported by the network

The algorithm appears to refer to a dynamic charging scheme, where charges not only vary from connection to connection, but also during a connection, but this is not explicitly stated

Comments

A definitive evaluation of this proposal cannot be made pending receipt of the expanded version of this paper, and will require greater familiarity with the theory of prices than I currently possess There are a number of ambiguities in the paper, so that several implementations of the proposed scheme, with different real-world performances, might be possible The authors state that theirs is a preliminary argument, and I would agree that their proposal requires further elaboration before its strengths and weaknesses can be identified

3 1 4 Conclusions

In our trial, the three-term algorithm is used The choice is based upon the judging criteria we listed in the first chapter

	Simplicity	Predictability	Fairness
Statistical Measurement Scheme	Need only bandwidth, easy to implement Could miss a burst, if interval not chosen carefully	Based on traffic statistics, can be hard to predict when source is bursty	The possibility of missing a burst can be used by malicious users, and thus be unfair for NO
Effective Bandwidth Scheme	Hard to measure effective bandwidth in hardware	Difficult to predict	Fair to user if prediction is chosen carefully Taken QoS into consideration
Effective Bandwidth Estimation	Can be implemented in several switch architecture	Hard to predict	Result might not be fair to users, due to simulation deviations
Three-Term Scheme	Easy to implement	Easy to predict if no duration term, or volume term is dependent on duration	Can be fair to users and NO's if parameters chosen carefully
Resource Pricing	Can be difficult to implement	Hard to predict	Time scale might be a problem

Table 3 1 Comparison for different charging schemes

Judging from the above table, we can see that three-term scheme is an obvious winner

3 2 Existing Internet Network Level Charging Mechanisms

The Internet is one of the largest packet switching networks in the world with nearly 20 million users located in 135 countries The Internet uses essentially a pricing policy based on the access [7]

For large organisations, users pay the membership fee of a regional network, which gives them an access to the network The access is unlimited and the fee is based on the bandwidth of the connection

For individual users, most Internet service providers charge the customers at a fixed rate, a bit similar to leased line charging The Internet user would normally pay at a fixed rate each month to the Internet service provider for unlimited Internet access, and at the same time, pay a phone bill to the telecommunications companies for the telephony usage There is no specification of limitation on bandwidth a user can take, which is normally limited by the user's modem or the bandwidth of the telephone line Amount of usage is normally not reflected in such a bill, and the charge might not be fair to the customer Consider the following scenario raised by Jeffrey K MacKie-Mason and Hal R Varian [6] Sally sends a packet that crowds out Elena's packet, Elena suffers delay, but Sally does not pay for the cost she imposes on Elena This scenario shows the other side of fixed rate charging apart from its simplicity

In China, Internet providers charge people by the amount of usage Users pay by the amount of information they receive, usually, per kilobytes Still the customer need to pay a separate phone bill, and the above scenario isn't solved, as Elena still needs to pay for the extra cost caused by network congestion, since duration and other QoS parameters are not included in these types of charges

3.3 Proposed Internet Charging Mechanisms

The work from two papers are discussed in this section, "Billing Users and Pricing for TCP"[2], and "Pricing the Internet"[6]

3 3 1 Pricing for TCP at Gateway Level

In [6] R J Edell et al discussed a network level charging mechanism for TCP/IP networks The billing system proposed in that paper is shown in the diagram below

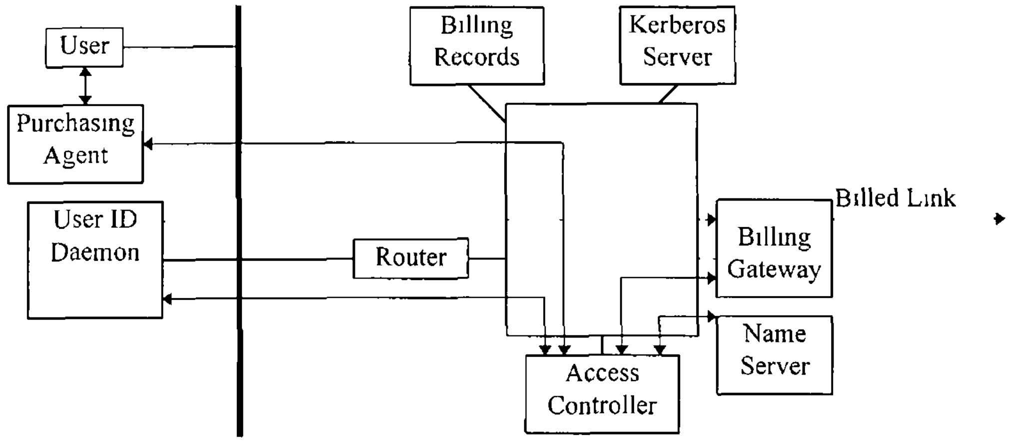


Fıg 3 4 Components of the bıllıng system showıng the major communıcatıons required to establısh a connectıon

The above dıagram shows a sıngle admınıstratıve domaın that ıs connected to the outsıde world vıa a bıllıng gateway The bıllıng system controls users' access to the bılled lınk by allowıng or dısallowıng TCP connectıons and by meterıng users' TCP traffic once a connectıon has been establıshed

The basıc operatıon ıs as follows when a user attempts to establısh a TCP connectıon wıth the outsıde world, the bıllıng gateway postpones the establıshment of the connectıon whıle ıt trıes to ıdentıfy the orıgınatıng user The system contacts the user, verıfyıng that they want to establısh the connectıon and that they are prepared to pay for ıt If the user accepts the connectıon, the normal connectıon establıshment ıs allowed to contınue and the bıllıng gateway wıll begın meterıng thıs connectıon's traffic

The system ın the paper dıscussed ıs for ınternal bıllıng, and only outgoıng traffic ıs monıtored and hence bılled However, the system can also be used for external bıllıng, such as bıllıng for World Wıde Web The complete system relıes on the performance of the bıllıng gateway, whıch ın the actual system configuratıon ıs a router As the authors claım ın the paper that thıs system can also be used to control congestıon by enforcıng a prıcıng scheme It ıs also poınted out ın the paper that users can hıde data ın TCP or IP header's optıon field Even though thıs can be tested by examınıng every TCP/IP packets, ıt ıs worth poıntıng out that for IP over ATM networks whıle cells are beıng transmıtted at a tremendously hıgh speed, thıs method mıght not work properly I e a gateway mıght cause serıous congestıon by examınıng all headers of each packet Most importantly, the paper mainly discussed how to use the configuration to charge users, not about how to charge and bill users, i e no actual charging scheme is given, only a charging system design is proposed

3 3 2 Cost Based Charging for Internet

In MacKie-Mason's paper [6] a cost based charging scheme is given The authors emphasise that users should face prices that reflect the resource costs that they generate so that they can make informed decisions about resource utilisation

The authors listed five costs

- *The incremental costs of sending extra packets* If the network is not congested, this is essentially zero
- *The social costs of delaying other users' packets when the network is congested* This is not directly a resource cost, but should be considered part of the social cost of a packet Users bear this cost through delay and dropped packets, and would often be willing to pay to reduce congestion
- *The fixed costs of providing the network infrastructure* This is the rent for the line, the cost of the routers and the salary for the support staff
- *The incremental costs of connecting to the network* Each new connection to the Internet involves costs for access lines and switching equipment
- *The cost to expand network capacity* This normally consists of adding new routers, new lines, and new staff

The third and fourth cost is the set-up fee and line rental respectively in the existing Internet charging and billing However the idea of introducing incremental cost of sending extra packets is very interesting It can be used for solving part of the network congestion problem The idea of putting extra charge on a user for delaying other users, however, might be hard for customers to accept Because first of all, users have no access to traffic control, and it is completely for the network to decide which packet should be delayed (It is very hard to implement in hardware to find out which user's packets are being delayed) And secondly, if the network has a very narrow bandwidth, it gets congested very easily, and this is completely the NP's responsibility

3.4 Internet Service Level Charging Techniques

In this section, we introduce two service level charging proposals both from IETF Internet draft [3][19] These two proposals are not charging schemes, but only ways of how to measure charging parameters

3 4 1 Incorporating HTML as Charging Tool

D E Eastlake 3^rd's paper [3] claims that application level Internet commerce requires a means to (1) indicate prices and acceptable methods of payment, (2) tender payment, and (3) issue a receipt acknowledging payment or indicate if payment fails

The paper introduced a complete system for billing users The charging information is embedded in "price tags", which contains amount of currency user should pay, unit of currency, such as US dollars, and Irish pounds It also contains the name of the payment system through which the currency should be paid When the user encounters a price tag, he is prompted by the system whether he wishes to pay such bill to receive the information, and in what currency and payment system he wishes to pay Once the user agreed to pay the required amount and selected a payment system, the system will acknowledge the receipt of payment

Such a system is very good for content charging, but is not dynamic The user can predict their bills by viewing the source code of the web page The scheme proposed here is very similar to the fixed charging method used in FIONN trial, which is discussed in chapter 6 in more detail An interesting feature of this proposal is that it can also be implemented on Telnet and SMTP protocol (email)

3 4 2 Internet Billing Using HTTP Extension Protocol

In [19], the authors claimed that on-line contract negotiation could also be obtained by extending HTTP protocol As we have introduced in the previous chapter that whenever a user request is generated for web browsing, an HTTP header is sent to the server, and the server processes this header by examining each field in it

A UPP protocol (Universal Payment Preamble) is implemented for this on-line payment system. Every time a user requests a page, instead of an HTTP header, a PEP header (HTTP Extension Protocol) is sent to the server which contains a UPP information. If the server supports UPP, it will then start a negotiation with the user, which allows the user to choose the type of currency, payment method etc. Once the payment is made, the user can then get the information he requested before. This method is quite similar to the one introduced in the previous section. The previous one is more general and can be implemented in other protocols, and this one is only designed for HTTP, i.e. web browsing. The two methods both require the server to have knowledge about charging beforehand and the user can always predict their bills.

3.4.3 Three-Term Charging Scheme

We have described Three-Term Scheme previously as a network level charging scheme. But it can also be used as a service level charging scheme.

$$C = D + E\times t + F\times v \qquad 3.18$$

The term D is a service fee, and its value depend on the importance of the information user is retrieving and the quality of service user is getting. The other two terms are same as before.

3.4.4 Conclusions

Again, the criteria listed in Chapter One is used, and a table is used to compare the service level schemes:

	Simplicity	Predictability	Fairness
Using HTML	Easy to implement, but HTML specific, can not be used if not using HTTP protocol	Easy to predict	Price is normally at fixed rate, and is not fair to users, and may not be fair to Service Providers either.
HTTP Extension	Easy to implement	Same as above	Same as above
Three-term Charging	Easy to implement	Relatively easy to predict, especially when network traffic is not busy, or duration term is	Scheme is adjustable, can be fair to both sides if carefully chosen.

		not used	

Table 3 2 Comparison for different service level charging schemes

It is quite clear from the above analysis, that all three schemes are quite easy to implement, and easy to predict The difference between the three schemes lies in the third criteria fairness The first two, due to implementation limitations, can offer only fixed rates, while the three-term one can offer a more flexible rate, depending on the type of information user has received, and the quality of service Again, it becomes a winner, and hence is chosen as service-level charging scheme in the field trial

Chapter 4. Investigation of Network Level Charging Scheme

4.1 Scenario

4 1 1 Introduction

This chapter investigates network level charging in a small office LAN The purpose of this chapter is not to develop charging schemes that can be used in commercial ATM networks, but to investigate how charges billed by the ATM network provider can be distributed to individual users in the LAN by a usage based charging method

4 1 2 Scenario

Let us imagine a corporation, with all its computers connected on an Ethernet LAN, or other type of LAN, with or without using any ATM links internally, but the corporation has an ATM link going outside the company, and all its external network activity happens on this ATM link The corporation will receive a bill from the ATM network provider at the end of each month or each year If all the sections or departments within this organisation are financially independent, in other words, they pay their own bills, then it is up to the organisation to try to divide the bill into small parts for each department, preferably usage based to be fair to the users The following diagram shows the above-described scenario

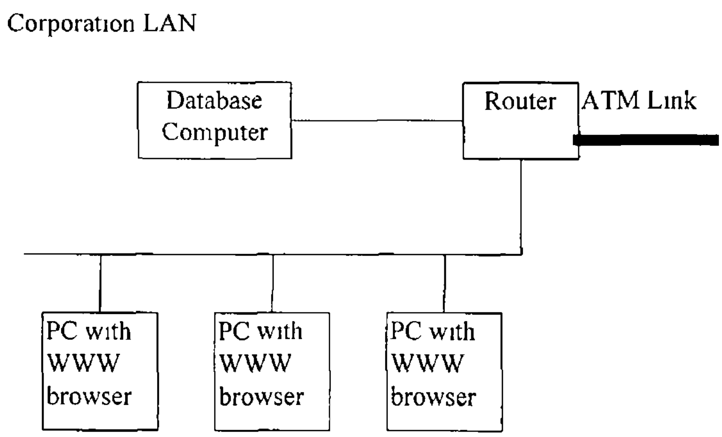


Fig 4 1 Scenario

The router in the above diagram is capable of recording network traffic on a host computer basis, and the database computer is used to store all the network traffic recorded by the router

In a future development, the charging and billing system will implement data collection and storage to a database, and users in the company with proper user authentication can browse the database by means of WWW pages The database at present is implemented so as to collect charging parameters for PBX telephony systems, and the purpose of this investigation is to recommend charging parameters and a database table structure for network level charging

4 2 Hardware and Software Specification

4 2 1 Hardware

The system consists of a CISCO router, and two PCs running Novell Netware IP/IPX, connected via ThinNet

The CISCO router is CISCO 2514 router, with 2 AUI Ethernet port, and 2 serial ports for connecting to WAN Also on the router, is a console port and auxiliary port, for users to issue commands or set-up the router The console port supports no hardware flow control, and transmits and receives data at 9,600 baud rate, with 8 data bits, no parity bit, and 2 stop bits The auxiliary port that runs a much higher baud rate of 38,000bps is used to connect to a remote modem, which has hardware flow control Both ports use RJ-45 connection, and RJ-45 to DB-9 and DB-25 adapters are supplied with the router

4 2 1 1 Hardware Set-up

It is best to use a graph to illustrate the network set-up

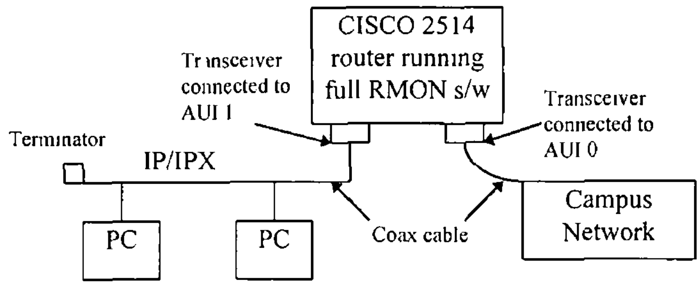


Fig 4 2 Network Set-up

The router's Ethernet ports use AUI 15 pin connectors, and the local area network on the campus is using ThinNet BNC connectors Two transceivers are used to make the conversion One Ethernet port (AUI 0) is connected to campus LAN, via a T piece, while the other is used for the small test LAN in the lab Two PCs are connected to this port (AUI 1), using bus topology, via coaxial cables The bus is terminated with 50Ω terminators on both ends T connectors are used to connect the PCs to the test LAN There are no cables between the T connector and the Ethernet card which is embedded in the PC, and both terminators are not directly connected to the T connectors at the end of the bus, but are extended at least 0 5 meters away from the T connectors, using coaxial cable

4 2 1 2 Router Configuration

The configuration of the router is done in two stages

1 Configuration using the set-up dialogue, and

2 Manual configuration

Set-up dialogue is used to configure the protocols, and SNMP groups The software supports IP/IPX/AppleTalk/DECnet protocols, but only IP, IPX and RIP have been enabled on the router, and all the other protocols are disabled Also enabled is the SNMP protocol, with the community string set to *public*

Two IP addresses are obtained for the IP protocol The first is the sub-network IP for the test LAN, which is designated 136 206 114 0 The other IP address is for the Ethernet port that connects to the campus network Its IP is 136 206 35 220 AUI 1 is assigned an IP of 136 206 114 254, as it is the port that connects to the test LAN 8 bits are assigned to subnet field on both Ethernet ports, and the subnet mask is automatically set to 255 255 255 0

Similarly, two IPX network addresses are also obtained, one is 88CE2301, which is assigned to the Ethernet port that connects to the campus network, and the other is 88CE7201, which is assigned to the other port which connects to the test LAN IP/IPX protocols are also enabled on both of the Ethernet interfaces

Finally, the router is assigned a name *flanker*, and terminal password for remote telnet login *vterm1* The passwords for enabling privileged mode are *bestplane, evenbetter*

Up to now, the first stage of the configuration is successfully finished, and if the user wishes to change the set-up using the set-up dialogue again, he could type *enable* to start privileged mode, and given the correct password The screen will show a different prompt, which indicates the user enters the privileged mode In this mode, the user can type in *setup* to re-start the set-up dialogue

The second stage is to use the manual configuration, since set-up dialogue provides only basic set-up options, and does not configure SNMP and RMON The configuration can be done by first entering the privileged mode using the command introduced above, and then typing *config term* The user will be asked to give one configuration command per line, and when finished, press Ctrl-Z If the user wishes to use the configuration when the router starts up, he should issue the command *copy running-config startup-config*

The commands that one should use to enable RMON on the interfaces are
First, type *interface ethernet 0,* to configure AUI port 0 The prompt will change to indicate that one is entering interface configuration Under the interface configuration mode, one should type either *rmon promiscuous* or *rmon native* to enable RMON on that port When done with the interface configuration, one should type *exit* to return to configuration mode RMON can be enabled on the other AUI port by issuing similar commands One only need to change *interface ethernet 0* to *interface ethernet 1*

The only difference between RMON promiscuous mode and RMON native mode is that when in promiscuous mode, RMON examines each packet that goes on the bus, while in native mode, RMON will only examine packets that are designated to this port. Obviously, the former will take much more CPU time for processing.

After RMON is enabled on both interfaces, one should use the command *rmon alarm* and *rmon events* to set-up a few RMON alarms and events to enable the other RMON groups. According to CISCO's technical documents, the other 6 RMON groups (except the statistics group) are not enabled unless the user sets up a few alarms and events groups. However, it is found out in actual set-up that one needs to set-up the control table in order to enable an RMON group. For example, to set-up the RMON host group, one needs to set the data source entry in the control group to e.g. "ifIndex 1", which means the group is set-up to monitor traffic on interface one, then to set the status entry to 1, which means the entry is valid. Only then, will a host entry appear in the RMON MIB. After the set-up of the control tables, the router will monitor the traffic on the network to add host computers into its tables.

4.2.2 Software

The software used in the trial consists of two parts:

1. Management software: provides the means of setting up and configuring the router.
2. Data collection software: used to collect traffic information from the SNMP MIB, and store this in database tables. The software also allows the user to retrieve and browse these tables, and can produce bills according to the traffic information.

4.2.2.1 Management Software and RMON Specification

The management software provided along with CISCO 2514 router allows the user to configure the network the first time the router is switched on. The user can type commands on the console machine, or can alternatively issue commands via a web page stored in the router, once it is set-up properly. The manual and also the CDs provided with the router explains in great detail how a router can be configured, and the usage of each command, along with sample output. As we are only interested in network traffic per host basis in this trial, only that part of the commands are of interest to us. These commands are RMON related, e g "show rmon history", "show rmon hosts", etc All these commands show a specified table in the RMON MIB The remaining part of this section will explain briefly what RMON is, and how it can be used in our experiment

RMON stands for Remote Network Monitoring It defines a remote monitoring Management Information Base (MIB) that supplements MIB-II in SNMP, and provides the network manager with vital information about the inter-network [20] RMON is capable of providing value added data for each individual device in the sub-network, unlike SNMP, which only provides network information that is local to the SNMP agent

4 2 2 1 1 SNMP

SNMP stands for Simple Network Management Protocol It defines a set of MIB (Management Information Base) data structures to store network related information It also provides a means of communication between an SNMP probe and an SNMP manager, including retrieving and setting of MIB elements [20] It was designed to be an application-level protocol that is part of the TCP/IP protocol suite, and is intended to operate over the User Datagram Protocol (UDP) As with any network management system, the foundation of a TCP/IP based network management system is a database containing information about the elements to be managed This database is referred to as a MIB SNMP MIB is a database structure in the form of a tree

MIB-II is the most important of the MIB specifications, covering a broad range of managed objects SNMP version 1 MIB contains 9 groups

- *system* This group provides general information about the managed system, including system up time, physical location of the node, etc
- *interfaces* It contains generic information about the physical interfaces of the entity, including configuration information and statistics on the events occurring at each interface Each interface is thought of as being attached to a sub-network It provides traffic statistics on each interface, such as speed, in and out byte count

- *at* Stands for Address Translation group Each row in the table corresponds to one of the physical interfaces of the system, and this table maps network addresses to physical addresses In the real set-up, this group stores all computers' addresses that are connected to the sub-network on the two interfaces This is useful when translating MAC address to IP address
- *ip* This group contains information relevant to the implementation and operation of IP at a node
- *icmp* ICMP is an integral part of the TCP/IP protocol suite, and a required companion to IP This group contains information relevant to the implementation of ICMP at a node
- *tcp, udp, egp* These are all protocol related groups, and provides no information about the network traffic

A close examination to SNMP MIB-II groups and objects shows that the MIB objects provide only information up to the interfaces, and does not provide and further information for stations connected on sub-networks (with the exception of the *at* group, which provides address translation of each station connected to the interfaces) This means SNMP does not provide adequate information related to charging, and this role, has to be fulfilled by the RMON groups

4 2 2 1 2 RMON

RMON is a MIB extension to MIB-II standards, which has 10 groups

- *statistics* maintains low-level utilisation and error statistics for each sub-network monitored by the agent, i e traffic information on the interfaces
- *history* records periodic statistical samples from information available in the statistics group
- *alarm* allows the management console user to set a sampling interval and alarm threshold for any counter or integer recorded by the RMON probe
- *host* contains counters for various types of traffic to and from hosts attached to the sub-network
- *hostTopN* contains sorted host statistics that report on the hosts that top a list based on some parameter in the host table

- *matrix* shows error and utilisation information in matrix form, so the operator can retrieve information for any pair of network addresses
- *filter* allows the monitor to observe packets that match a filter (The monitor may capture all packets that pass the filter or simply record statistics based on such packets)
- *capture* governs how data is sent to a management console
- *event* gives a table of all events generated by the RMON probe
- *tokenRing* maintains statistics and configuration information for token ring sub-networks

This experiment is to investigate how an overall bill by an ATM network provider can be divided among all the terminals using usage based charging, so those groups that are concerned primarily with the collection of inter-network traffic statistics are of interest to us, e g *statistics, history, host and hostTopN* Of all these groups, the most interesting one is probably *host* group, which provides information about traffic statistics on each host machine that is connected to sub-network linking with the router's interfaces

It was found out that the RMON groups can only provide traffic information up to the interfaces, and can not go beyond This is an undesirable feature, since a college might have more than one router and it is best that one can use RMON at the access router to get traffic information, instead of having to go to each router and retrieve various information for billing Retrieving information from the access router only is also desirable, because it can be set-up such that the router will not record internal traffic, which will not be charged by network operator But this internal traffic will definitely be recorded by RMON As RMON II claims that it can record network traffic at application level, and categorise different traffic by different types of protocols, it might also provide the functionality that we want But since RMON II is not available, the research is centred on Cisco's private MIB, the IP accounting MIB This MIB records source and destination IP addresses, and also byte count between the source and destination IP accounting could be set-up on the router interfaces such that only traffic going in one direction will be recorded RMON MIB is also discussed here because it is possible that future RMON standards will include the functionality provided by Cisco's MIB, and we do not wish to limit this study to Cisco's router only

4 2 2 2 Data Collection Software

The data collection software is written in Microsoft Visual C++, which provides database connection and SNMP functionality The SNMP functionality is fulfilled using HP's object oriented SNMP library, SNMP++ version 2 5 The database we have chosen to use is Microsoft's Access database The reason why it is chosen is because we have only the ODBC driver for Access

4.3 Existing IP Analysis Tool Comparisons

There are some commercial Network Management Systems (NMS) available on the market, and this section is to compare some of these products' functionality to examine their fitness in network charging

The product list is obtained from [21] The author listed and examined sixteen vendors and their Internet analysis products The following table gives a simplified list of the products showing the vendor, product name, collection source, and their prices

Vendor	Product	Collection Source	Price
Bay Networks Inc	Optivity Enterprise 7 0	RMON, RMON II, SNMP MIB I and II agents	Optivity $5,995 to $17 995 four port Ethernet probe $7 500
BGS Systems Inc	Bestview	RMON II SNMP MIB I and II agents	$24 500 console only
Computer Associates International Inc	Application Response Manager	SNMP I and II proprietary agents	$2 500 1 agent and console
Compuware Corp	Ecoscope	proprietary probe	$36 000 10 probes and console
Concord Communications Inc	Network Health LAN/WAN	RMON I and II agents SNMP proprietary collection agent	Console $15,000 Weblink $10 000 Probes $250 per segment
Frontier Software Development Inc	Netscout Manager	RMON RMON II	Windows console $3 995 Unix console $5 995 four interface WAN probe $11 995 Ethernet probe $6 995
Hewlett Packard Netmetrix Division	Netmetrix Internetwork	Proprietary IRM probes RMON and RMON II probes	HP Quad Ethernet LAN probes start at $6 000 Internetwork Response Agent $495 Netmetrix Internetwork Monitor console $1 495
Jyra Research Inc	Midlevel Manager	RMON, SNMP proprietary agent	$6 500 console only
LANquest Group	Interview	Proprietary and RMON probes	Console with 8 sentry agents $9,900 Sentry pack of 8 $1 995
Network General Corp	Distributed Sniffer 5 0	Proprietary analysers	DSS server %5 995 Windows console $3 995
Optimal Networks Corp	Optimal Network Monitor 1 2	Proprietary agents	$2,500 1 agent and console
Platinum Technology Inc	Wiretap	Proprietary probes	Console $8 000 probes $2 000 each
3Com Corp	Transcend Traffix Manager	RMON II SNMP agents	Transcend monitor starts at $3 295(for dual port Ethernet) Traffix Manager $7 995
Technically Elite Inc	Meterware for Windows 95/NT	RMON I and II SNMP agents	Ethernet probe $2 595 Meterware Application $2 995
Tinwald Networking Technologies Inc	Internet Snapshot	Proprietary agents	$395 1 agent and console
Visual Networks Inc	Visual Onramp	Proprietary agents	$2 995, 1 agent and console

Table 4 1 List of IP Traffic Analysis Tools

None of these tools are designed for accounting purposes, but just for analysis of network traffic, and monitoring, mainly for network maintenance However, since they provide traffic information, they should also provide some charging related information, which can be used for office charging and billing

It is clear from the above table that six out of the listed products use proprietary MIBs only to analyse network traffic, which means that as time varies, these MIBs can be changed and their older version no longer compatible with the newer ones User would have to replace the existing software Good accounting software though, should conform to existing standards, and utilise standard MIBs such as RMON, SNMP to collect traffic related information Thus, even when new versions of the standards are released, the users of the software would need very little or no change at all to his existing system One other product from Computer Associates International Inc , uses a proprietary MIB as well as SNMP MIB As we discussed in the above section, SNMP MIBs provide only information not beyond the interface that the SNMP agent sits, and does not give the user a global view of traffic information over the sub-network To obtain the global information using SNMP MIB, the user has to use SNMP probes at all the interfaces that he wants to monitor, which is not economical

From the above analysis, we can conclude that the seven products we pointed out are not suitable for use as office billing systems as they either lack of means to monitor traffic per host basis, or do not use standard functionalities

Of the remaining nine products, only one of them provides usage based cost estimation, that is the product from Frontier Software Development Inc , Netscout Manager, which also provides the user with a web based interface, at a price of about $4,000 The other software all concentrates on traffic managing and statistical trending, but fails to put any emphasis on charging and billing

Another very useful feature of Netscout Manager is its ability to communicate with a SQL database, and store retrieved information into the database The vendor also claims that it allows the user to make queries to the database to retrieve information he wants, which makes it a very desirable front end for a billing system However, the high price will make it less attractive to customers if used merely as billing software

What's missing from the list but well worth mentioning are two products, one from CISCO, named Traffic Director, and another from Castle Rock, named SNMPc

The former is a similar product to Netscout Manager from Frontier, and in fact, CISCO has announced a strategic alliance with Frontier, which designates the company as its primary vendor of RMON based technologies From the description of the two products, they both support CISCO's hardware products, e g routers and switches, and both provide similar functionality, such as database storage and query Again, priced at a very high $7,000 it is unlikely that a user will choose to use it only as an accounting tool, even though it provides the functionality

The latter is priced at less than $1,000, but is an SNMP & RMON front-end tool, rather than a network management tool or billing system It does not provide database connection or web connection, however, it does allow the user to set-up RMON and SNMP MIBs, and retrieve information from the MIBs The software also supports on-line polling, and the user can save a snapshot of the MIB into a disk file Automatic MIB information logging is not supported in the software

From the analysis of all the traffic analytic tools available from the market, we can reach the conclusion that current traffic analysis tools do not place emphasis on usage based cost analysis They are either priced at a very high price, which will normally only be accepted by large companies or universities, or use proprietary probing methods, which might make software/hardware upgrade more difficult for the end user

4 4 Trial Set-up

The system was set-up as shown in Fig 4 2, with two PC's connected to the small LAN used for the trial Both computers are 486 DX2 66Mhz, one running Windows 95 and the other Windows 3 11 FTP is used on both computers to create out-going network traffic Thirty-one sets of traffic data were recorded with thirty of these being used as network traffic data, while the final set was used as reference data Because the counter that records the number of bytes transferred from source IP to destination IP records the accumulative values only I e number of bytes transferred from source to destination from the time the source destination pair appears in the accounting table, till the last time of transaction, a reference value will have to be used to eliminate previous traffic

The interval for retrieving data from SNMP MIB is set at 4 minutes, and every four minutes, network traffic was created manually using FTP to transfer several randomly chosen files The data collection software was set to automatically fetch data from the router every four minutes, until it collected thirty-one sets of data The data were later converted manually using the collection software into a tariff table, which consists of the source IP address of the traffic, the net amount of traffic generated by the source IP, and retrieval time and date of the traffic After the conversion is done, the tariff table is then fed into the charging and billing system in the data collection software for further analysis

4.5 Charging Schemes

Several charging methods are used and compared to find the most suitable method Even though the LAN is connected to an ATM link, and is charged by the ATM Network Provider for the network usage, the internal billing system can only record traffic on the LAN, rather than the ATM network None of the ATM charging methods introduced in chapter 3 was used In this trial, only network usage is used as charging parameters Variation of usage based charging methods are applied to find one that is

1 Fair to the user
2 If possible, informs the user of how much he's already spent by the time the user checks his usage of the network
3 If possible, predict how much the user is going to pay by the end of month if his pattern of network usage does not change

4 5 1 Simple Usage Based Charging Method

Simple usage based charging is our first choice

At the end of one month, find out how many bytes a user transferred from his computer onto the Internet Suppose for each user i, the byte count is t_i Then the total amount of traffic generated is

$$TotalTraffic = \sum_{i=1}^{N} t_i \qquad 4\,1$$

Where N stands for the maximum number of users

The charge c_i apportioned to user i at the end of the month using simple usage based charging is then

$$c_i = C \times \frac{t_i}{TotalTraffic} \qquad 4\,2$$

where C is the total charge for the network usage to the company for this month

4 5 2 Daily Usage Based Charging Method

The second method applied is called daily usage based charging It is similar to the first one, except that it assumes the network is charged equally everyday by the network provider, i e ,

daily total charge = monthly total charge / number of days in the month

Then apply simple usage based charging method to everyday for each user, and then the charge up to the present day for one user is

$$c_i = \sum_{j=1}^{today} \frac{t_{i,j}}{TotalTraffic} \times DailyTotal \qquad 4\,3$$

where *TotalTraffic* is total usage on a particular day j, *DailyTotal* is daily total charge defined above, and $t_{i,j}$ is traffic produced by user i on a particular day j

It is quite obvious that simple usage based charging is very fair to the user, as it only checks by the end of month how many bytes a user has transferred, and make no other assumptions The disadvantage is that this method can not predict how much the user is going to pay for the network traffic he used Also, there is no way the user could know how much he needs to pay for the traffic that he already used until the total is computed at the end of the month As a comparison, the daily usage based charging method does tell the user how much he has to pay for the traffic he already used, but does not give indication of how much he is going to pay at the end of the month Most importantly, this method is not quite fair to the user Assume that one user comes in on a Sunday afternoon, and sends out an email, where no one else is in the company using the network Then on this particular day, no matter how short the email might be, this user is going to pay for the entire bill on this day, since there is no other traffic

That's why we also introduced two other methods, trying to achieve all three expectations we've listed above for our charging and billing system

4 5 3 Traffic Interpolation Method

The first one of the two is called traffic interpolation method Because the byte counter's value always increases, we can say that if the value is plotted everyday, we should see an upward trend in the network traffic If we use least mean squared method to find the best straight line that fits all the traffic points recorded, then we can predict how many bytes the user is probably going to use by the end of the month, and hence, predict how much he is going to pay When it reaches the end of the month, then we can use simple usage based charging method to produce the actual charge to each user The method can be described mathematically as follows

Assume $t_{i,j}$ is the number of bytes user i used up to the day j, then to day N, we have a set of values, $t_{i,1}, t_{i,2}, \ldots t_{i,N}$, for all the users For each user i, we can then find a set of numbers a_i, and b_i, where the equation

$$t_{i,x} = a_i \times x + b_i \qquad 4\ 4$$

is the equation for a straight line that best fits the points $(1, t_{i,1}), (2, t_{i,2}), \ldots (N, t_{i,N})$ We can then put x = number of days in the month in the above equation for the straight line to find out the amount of traffic user i is going to generate at the end of the month Simple usage based charging method is then applied very easily to give prediction of charge to each user at the end of the month This method is very fair to the user, as at the end of the month, simple usage based charging is applied Also it can give user prediction of how much he is going to pay for his network usage, but can not tell him how much he needs to pay for his usage up the present day

4 5 4 Combination Method

The last charging method we use combines the advantages of the above three methods, and accomplishes all the three goals we set for the charging and billing system at the beginning of this section

To be fair to the user, simple usage based charging scheme is used at the end of each month to calculate user payment for his network usage To be able to inform the user how much he needs to pay for his current network usage, a modified daily usage based charging method is applied which will be discussed later in this sub-section To be able to predict as close as possible how much a user is going to pay at the end of month for his network usage, several methods are tried

The modified daily usage based charging method can be described mathematically as follows

Assume the payment for the company's use of the network is the same everyday, which is c_d, then obviously, $c_d = C / \textit{number of days in a month}$, where C is the total charge for the network for this month Then for the past N days, the company's expense on the network would be $N \times c_d$ Assume that the accumulative network usage for a user i up to the day N is t_i, then the amount $c_{Ti,N}$ that the user i has to pay for the past N days is given as

$$c_{Ti,N} = N \times c_d \times \frac{t_i}{\sum_{i=1}^{n} t_i} \qquad 4\ 5$$

where n is the total number of users At the end of month, this equation would become the same as equation 4 2

There are several options available to accomplish the task of predicting a user's payment at the end of the month

- We can use linear interpolation on traffic described in section 4 5 3
- For each user i, we have a set of $c_{Ti,N}$ for various N, and hence, we can do a linear interpolation on the set of $c_{Ti,N}$ for a given i Then we can let N be the last day of the month, which will give us a predicted value of the end of month payment for user i

- Or alternatively, we can do a linear scaling on $c_{Ti,N}$ when N is very close to the end of the month, i e the end of month payment C_i for user i would be $C_i = c_{Ti,N} \times (number\ of\ days\ in\ the\ month)/N$

The next section will give detailed analysis on all the three methods for bursty and non-bursty traffic situations

4.6 Billing

In the trial, we assumed that the monthly charge for network usage is at a fixed priced of 30,000 units, and there are 30 days in the month, which results in 1,000 units per day In later part of this section, we shall see that the charging scheme used by the NP does not have to be fixed rate method The resultant bill for the whole month given by the first three methods described in the last section, simple usage based, daily usage based and traffic interpolation method is given in the following table

User	Simple Usage Based	Daily Usage Based	Traffic Interpolation
1	14441 97	11846 51	14441 97
2	15558 03	18153 49	15558 03

Table 4 2 Bill comparison for different charging methods

It is quite obvious that daily usage based charging scheme gives a value quite far away from the simple usage based charging method Experiments have also been carried out to test if traffic interpolation method can correctly predict a user's monthly cost Data are listed here on bill produced for the first seven days and first twenty-four days for the same network traffic Bill for daily usage based charging method is also given as a comparison, but the value is how much the user has to pay for network usage up to that day, instead of monthly payment given in the traffic interpolation column

User	Daily Usage Based	Traffic Interpolation
1	1957 19	7330 52
2	5042 81	22669 48

Table 4 3 Bill for the first seven days

User	Daily Usage Based	Traffic Interpolation
1	9073 81	13755 46

2	14926 19	16244 54

Table 4 4 Bill for the first twenty-four days

As we can see, the predicted monthly payment given by traffic interpolation gets closer to the actual payment value where there is more traffic data

Experiments were then carried out to examine the performance of the three alternative prediction methods described in section 4 5 4

First, the same network traffic data used above was used as non-bursty network traffic data

Method	User	N=29days	N=28 days	N=27days	N=12days
$C_{Ti\,N}$	1	14276 08	13799 08	13306 25	2178 74
	2	14723 92	14200 92	13693 75	9821 26
Scaling of $C_{Ti,N}$	1	14768 36	14784 73	14784 72	5446 85
	2	15231 64	15215 27	15215 28	24553 15
Interpolation of $C_{Ti\,N}$	1	14107 44	14023 72	14023 72	6780 56
	2	16616 08	16777 02	16777 02	27242 29
Traffic Interpolation	1	14247 55	14179 38	14179 38	6016 62
	2	15752 45	15820 62	15820 62	23983 38

Table 4 5 Payment prediction for the past 29, 28, 27 and 12 days for non-bursty network

Amongst all three methods, it is quite clear that interpolation on current payment gives the worst prediction on end of month user payment, and the sum of all predictions for different users does not equal to the total due monthly charge This is because interpolation on payment only tries to find a best straight line that fits all the payment points, and when the user wants the prediction, gives the point on this straight line as the predicted value It is totally dependent on traffic pattern, and nothing else Of the other two methods, scaling of payment and traffic interpolation, we can see that they both give very close results, while traffic interpolation method produces slightly better results

The traffic data is then modified such that user 2 has no traffic from day 18 to day 21, and from day 22 to day 28 The traffic for both users at the end of the month, however, remains the same, so that the monthly bill for each user will still be the same using simple usage based charging scheme, and only the predicted values will vary Below is the table we get for the past 29, 27 and 22 days for the two users using the three prediction methods

Method	User	N=29 days	N=27 days	N=22 days
$C_{Ti}N$	1	14276 08	13799 08	9542 46
	2	14723 92	14200 92	12457 54
Scalıng of $C_{Ti}N$	1	14768 36	14784 72	13012 45
	2	15231 64	15215 28	16987 55
Interpolatıon of $C_{Ti}N$	1	13774 15	13523 99	12780 69
	2	16949 37	17360 01	18640 85
Traffic Interpolatıon	1	13886 49	13666 12	13004 16
	2	16113 51	16333 88	16995 84

Table 4 6 Payment predıctıon for the past 29, 27 and 22 days for bursty network

Agaın, payment ınterpolatıon gıves the worst result, but thıs tıme, traffic ınterpolatıon only gıves a slıghtly better predıctıon value than payment ınterpolatıon, because of the burstıness of the traffic Scalıng of payment on the other hand, gıves much better predıctıon values, except at day 22, when ıt gıves values very close to those gıven by traffic ınterpolatıon method

A graphıc comparıson for the three methods can show the dıfferences between the methods more clearly

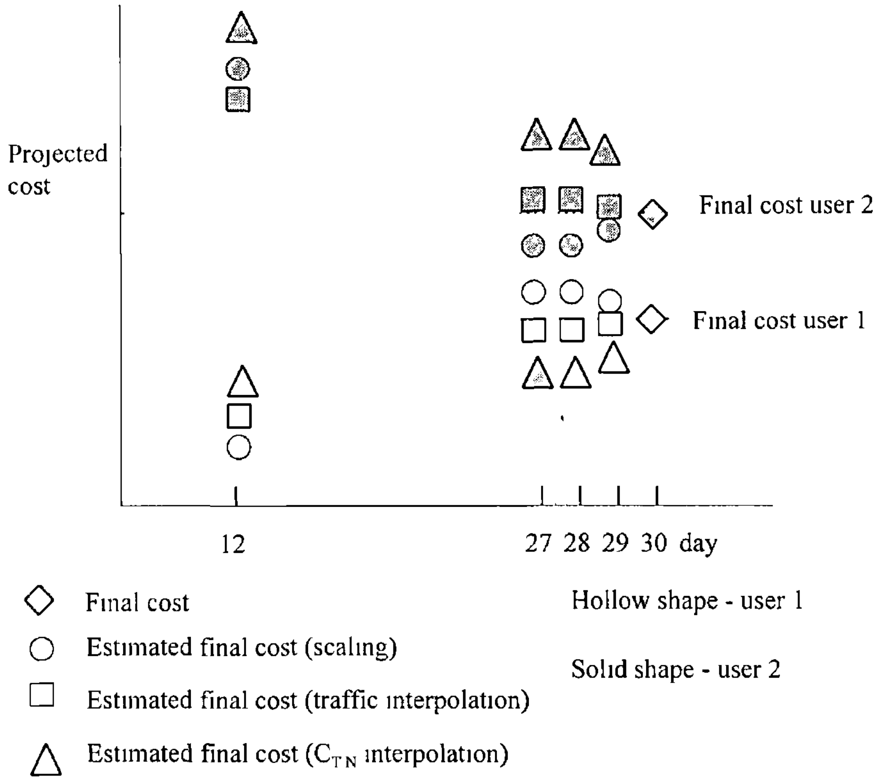


Fig 4 3 Illustration of cost comparison for the three methods for non-bursty traffic

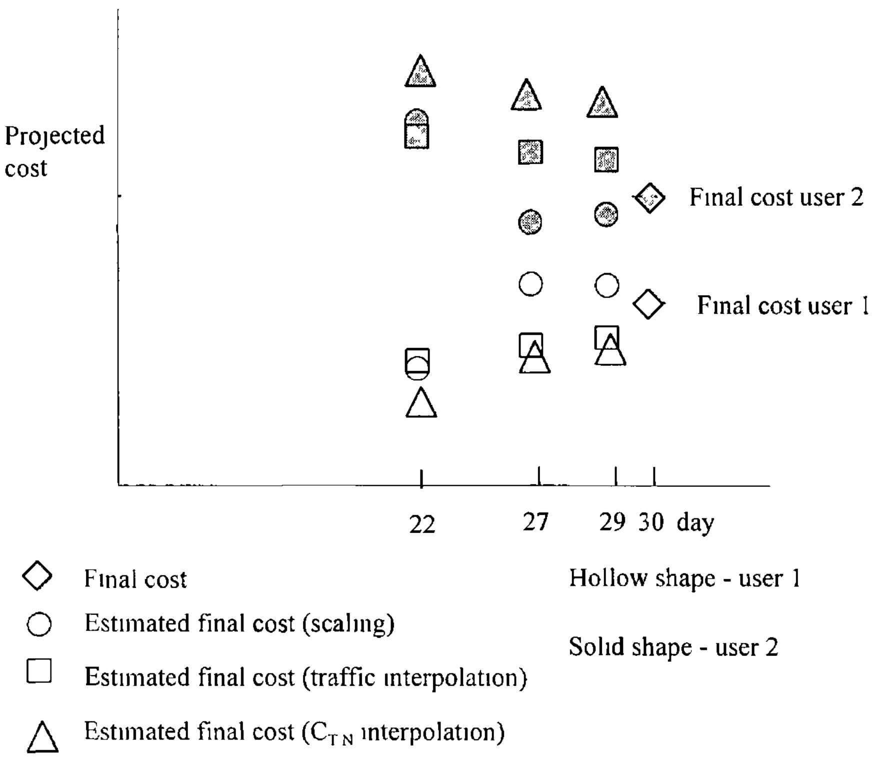


Fig 4 4 Illustration of cost comparison for the three methods for bursty traffic

The analysis for bursty and non-bursty network traffic shows that using traffic interpolation to predict user's expense will be quite inaccurate when the network traffic is very bursty, while using linear scaling of $C_{T1,N}$ when N approaches the end of the month could give better results Even when the traffic is quite smooth, scaling directly could give result close enough to real values This suggests that when predicting end of month payment of a user when it is almost near the end of a month, scaling should be given more emphasis At the beginning of the month, when scaling method does not make much sense, traffic interpolation should be given more emphasis, and while in the middle, each set should be taken into account more or less equally Based on this idea, a month is divided into three sections, with 10 days per section For the first section, prediction produced by traffic interpolation is given 80% weighting, while 20% is given to scaling method For the second section, weighting is assigned 50%, 50% While for the last section, scaling is given 80% weighting, and traffic interpolation is given 20% weighting, thus we can achieve all our three goals

- Very fair to the user, as the final bill to the user is simply based on his/her usage of the network, i e user bills are proportional to one's amount of traffic
- Gives user idea how much he/she has to pay already for his/her network usage, and as the day approaches end of the month, this value closes in to the value of the monthly bill to the user
- Gives the best prediction on user expense possible for all days in the month

The reason that interpolation on $C_{T1,N}$ is not used at all is because the sum of all user expenses does not equal to the total monthly charge, while is hard to give a weight Proof of correctness of weighting method is given here

Suppose the predicted monthly charge for all users using scaling is a_1, a_2, a_n, where there are n users, and monthly charging using traffic interpolation for each user is b_1, b_2, b_n Also assume total monthly charge to the network is M, then we have

$\sum_{i=1}^{n} a_i = M$ and $\sum_{i=1}^{n} b_i = M$ If scaling is given weighting of p, then the weighting given to traffic interpolation is *(1-p)* We can then combine the two equations into one using different weighting

$$p\sum_{i=1}^{n} a_i + (1-p)\sum_{i=1}^{n} b_i = p \times M + (1-p) \times M$$

which can be re-arranged as

$$\sum_{i=1}^{n} \{p \times a_i + (1-p) \times b_i\} = M$$

where $\{p \times a_i + (1-p) \times b_i\}$ is the weighted monthly expense for user i, and we can see that the sum of monthly expense of all users equals to total monthly expense, which proved this method is correct mathematically The bill for the same network traffic is listed below

User	29 days	28 days	27 days	25 days	22 days	18 days	17 days	16 days	14 days	10 days	8 days
1	14689	14690	14674	14637	14565	13490	13131	12553	11007	6111	6890
2	15311	15310	15326	15363	15435	16510	16869	17447	18993	23889	23110

Table 4 7 Predicted monthly bill for users, rounded to the nearest integer

Compared to the values given in the other tables, we can see that the improvement in the predicted value is obvious The first five sets of data is very close to the actual monthly expense of the user (within 2%), while even though the values for 18 days and less looked quite far away from the actual values, the improvement is still quite significant (compared to 9488 25 and 20511 75 using traffic interpolation for 15 days, the values for 14 days in this table are much closer to the actual values) From the comparison, we can reach the conclusion that using this method does give better predictions on user expense It is hard though, to choose a fixed set of weightings, as this will depend on not only a single user's traffic pattern, but also on other users' network traffic Hence, the weightings could only be estimated, and probably be set by managers, when the company has certain knowledge of its network traffic pattern

In the above measurement, we assumed that the charging scheme used by the NP is leased line charging scheme, but it does not have to be true For example, the NP could also use the three term method that we choose for the other field trial described in chapter 6 The duration measured by an ATM meter might be a little bit longer than that measured by the LAN router, depending on the protocols used in the ATM network But the duration will still be more or less the same The LAN manager can use his measurement of time interval to approximate the measurement of the ATM NP The LAN manager can also predict the volume term because the ATM layer only adds 5 bytes header to every 48 octets The LAN manager should be able to find out the actual traffic recorded by the ATM network very easily by checking his own byte count

If some other network charging scheme is used by the NP, this prediction system might not work But as long as the LAN manager can measure the parameters used by the NP, he should be able to predict his bill (parameters like peak bit rate and mean bit rate can be easily measured in a switch, and be converted to peak cell rate and mean cell rate) Thus the algorithms described above could still be used to distribute network charge

Chapter 5. Implementation of Service Level Charging System

5 1 Introduction to Service Level Charging

In order to investigate service level charging schemes, a World Wide Web server is modified to be the charging system The purpose for designing such a system is to examine how IP over ATM should be charged, and to investigate service level charging schemes The graphical representation of the scenario is given in the following diagram

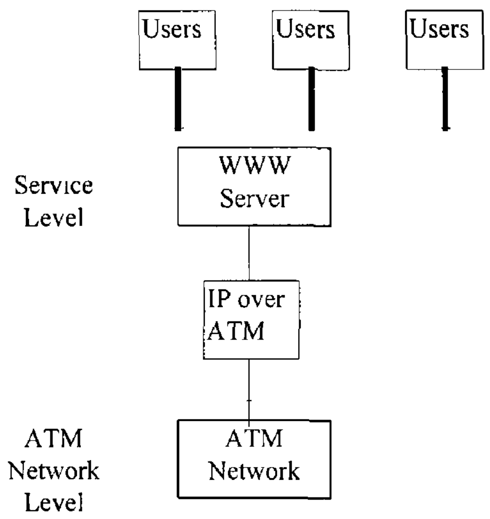


Fig 5 1 Structure of WWW service over ATM network

As in most cases, the SP is not the network provider (NP), and hence information for network level traffic available to the NP is not normally available to the SP Hence in order to conduct service level charging, the WWW server would have to be able to provide charging parameters to the service provider (SP)

Most WWW servers available from the public domain use common log format for their log files, which will be explained in later sections The common log format log file does not, however provide enough service level charging information, hence either a new WWW server should be designed from scratch or an existing server should be chosen and modified to meet the requirements of the field trial The latter will obviously take much less time and effort

5.2 Selection of WWW Server

To choose a server that can be modified easily to meet charging and billing requirements, a list of all available servers are drawn, with all their features and requirements listed The complete list of servers is obtained from W3 work group web page at http //www w3 org/ in September 1995

Server Name	Language	Platform	Public Domain
Apache	C	UNIX	Yes
CERN	C	UNIX, VMS	Yes
NCSA	C	UNIX	Yes
Spinner			GPL License
Alibaba		Windows NT	
CL-HTTP	Common Lisp	UNIX, Mac, PC	
GoServe		OS/2	
GN	C	UNIX	
GWHIS Server			No
HTTPS		Windows NT	
Jungle	Tk/TCL		
MacHTTP		Mac	
Netsite			No
Perl Server	Perl	UNIX	Yes
Phttpd Server	C	SunOS 5 4	Yes
Plexus	Perl	UNIX	Yes
Purveyor		Windows NT, '95	
REXX		VMS	
VAX/VMS Server		VMS	
SerWeb		Windows 3 1	
Website		Windows NT, '95	
Windows HTTPD 1 4		Windows 3 x	

Table 5 1 List of World Wide Web servers

Amongst all the possible choices, NCSA's WWW server is chosen to be our candidate for the following reasons

- Written in C, quite efficient
- Easy to install, installation manual available on web page
- The only machine that has outside Internet access is a UNIX machine
- Supports common format log file

Another very attractive feature about NCSA server is its user authentication This is only important if the web page maintainer wants to provide password to his files, but we will show that this feature is quite important for service level charging in chapter 6

5 3 NCSA WWW Server Structure Analysis

To modify the NCSA server, its structure was first analysed The server consists of three major parts

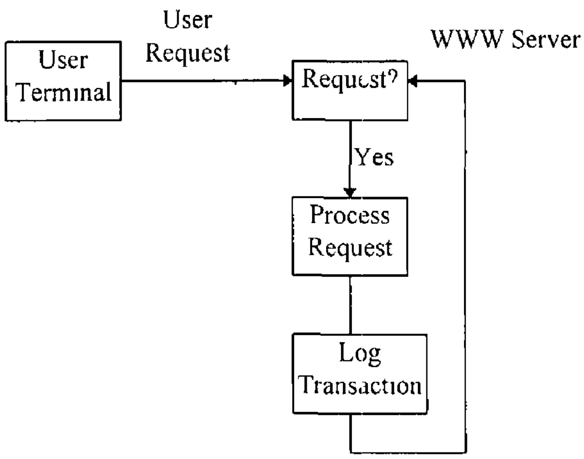


Fig 5 2 WWW Server Structure

The first part is used to establish connection between the World Wide Web server and the user The server would start up by initialising several sockets as its output and set them into listen mode This sets the sockets to await for the user request Such a request is generated by the web browser at the user terminal when the user keys in a URL e g "http //www w3 org/" and hits the RETURN key The browser at the user end will then start another socket and look for the server's socket at a specified port number (default is 80) The HTTP request header[8] will then be sent along this socket to the server end

Once the server socket detects such a message sent down the pipe, it will then switch itself into receiving mode A connection between the user and the server is then established

In the second part, the server passes the received user request to a function named *process_request()* This function will extract the request method from the HTTP header, and send them to corresponding functions (The standard methods are POST, GET and HEAD) The functions that receive the rest of the header information will then extract the name of user requested file, such as "index html", and locate the requested file in the World Wide Web server system The file then will be read into a string buffer, and this buffer is then written to the socket that connects the user and the server The user's browser will receive this file from the connection, and once all information in the file is sent, the socket connection is terminated from the user end While the server is transmitting the file along the socket, a byte counter is set up in the WWW server to record the number of bytes sent

If a particular file access requires user authentication, the server will then check for the password file in the system specification, decrypts the password, and prompts the user to enter his or her user name and password Once the check is passed, the server will then send the requested file This authentication is done once every browser session, i e the first time user uses Netscape to browse a particular password protected page, he will be asked for the password, but as long as the user doesn't close this Netscape application, he will not be asked for the password no matter how many times he leaves this particular page

When the transaction is finished, the server then will log all the transaction information by calling another function *log_transaction()*, which is done in the third part of the server The time when the transaction is finished will be generated in this function, as well as number of bytes sent down through the socket, which is calculated during the file transfer process If the call is interrupted by either the network or the user, an error log will also be generated with information recorded A reference log is also created which records the name of all the image files and included files in one user requested HTML file, e g if a user requests index html, it might contain 10 GIF files, 2 JPEG files, and 1 sound file This reference file would show that all the 13 separate accesses recorded in the server log file (by the 10 GIF, 2 JPEG and 1 sound file) are generated by access to this index file

5.4 Modification to NCSA WWW Server

To investigate charging for ATM networks, WWW browsing information related to charging are required, e g distances, time duration, amount of information transferred, and importance of the information A full list of possibly required charging parameters is given in Appendix E (both network level and service level)

5 4 1 Available Charging Parameters

A careful examination of the table tells us that some of the parameters that might be useful for network level charging is not available in service level charging For instance, peak cell rate and effective bandwidth

As mentioned in the previous section, NCSA's server supports common log format file, which is commonly adopted by most of the web servers The common log format file looks like

arctis-ext wise edt ericsson se - - [31/Jan/1997 09 40 12 +0000] "GET /gif/china gif HTTP/1 0" 200 872

The first item is the user's IP address or IP number, second and third field is normally recorded as dash, as in most cases, these two fields are empty However, if implemented, these two fields should show the user's email address and user name respectively The time stamp is the time when the server finished sending the file and started logging The "+0000" field shows server's time zone offset to GMT A "+0000" means GMT The three entries in the quotation mark represent the request method, in this case, GET, the file transferred, in this case "/gif/china gif", and the HTTP protocol version The second last entry is a status number For a complete transaction, this number is always 200, and the last entry shows the number of bytes the server transferred

What we have in this common log format file that is useful as charging information is the time stamp, which might be useful for duration charging, the file name, for content charging, and the amount transferred, for volume charging The IP address can be used as user identification, although this is normally not sufficient

5 4 2 Modification to WWW Server to Obtain Other Charging Parameters

From the analysis above, we can clearly see that in order to gather enough service level charging information, we need at least another time stamp, that is at the beginning of each transaction, to give us the duration This is very easy to implement, as we've pointed out in the structural analysis of NCSA's World Wide Web server that all the user requests are received by a function named *process_request*, and this function then analyses user request A timing function is then programmed at the very beginning of this function so that the server also records the starting time of a transaction

In practice however, we found this way of programming produces zero duration for most of the transactions This could only mean that the WWW server sends out files at a very high speed such that most of the transaction is done within a second This means a timing function with greater precision should be used One reason that might have caused the high speed is because all the initial test is done either on the same machine that the WWW server resides or on a LAN which is of very high speed However, we should bear in mind that ATM networks operate at a very high speed, and the same situation might occur if the timing function is not replaced

After careful examination of the C function library, the only function that could record time in milliseconds *ftime* was selected Microsoft Visual C++ claims the function is compliant with ANSI C standard, and SunOS 4 1 3 manual says this function is supported by 4 3 BSD UNIX system Also claimed in the Sun Solaris manual that this function is supported by the operating system However, this caused a compile time error, when the server is to be installed on a Solaris machine in the field trial When this function is chosen instead of the original timing function, transaction time is clearly shown in the log file, normally in the range of a few milliseconds to about 50 milliseconds for text files and most of the image files A list of the modified functions is given in Appendix D

5 4 3 WWW Server Log File Format

An example of the new web server log file entry is given below

136 206 36 5 [23/Sep/1996 18 30 15 014 - 23/Sep/1996 18 30 19 515] "/iss08/image s/chinaexp/mosqbird jpg" 47819

The first item gives the user's IP number, while the second and third give the starting and finishing time of the transaction respectively The time is accurate to number of milliseconds In this case, we see that the duration is about 4 5 seconds, for a moderately small image file The time is much longer than what we had in the initial test stage (this data is obtained in field trial discussed in the next chapter) because the access is from Dublin to Haninge in Sweden, the media is not completely ATM network, but mostly ordinary Internet TCP/IP connection (only the part in Telia's laboratory is ATM network, as will be shown in the next chapter) The fourth item is the name of the file requested, in quotation marks, and the last item is the byte count of the file transferred

The reason that the user's address is given in IP numbers, rather than IP addresses is that there is no Domain Name Server (DNS) running on the gateway machine to the WWW server in the field trial

5 4 4 Other Modification to the Server

A UNIX domain socket was also added to provide the user on-line monitoring capability A test program was written to create this monitoring socket This program was run in the background as the server The socket created was set to listen mode to wait passively for client sockets to connect to it The socket implemented in the *log_transaction* function in the server starts a socket as a client, and connects to the socket server When a connection is established successfully, the World Wide Web server will send the logged information to the socket server program By receiving this information, the socket server program will process the data, and display the user's IP address, name of accessed file, duration and amount of information obtained on the monitor screen This socket connection is implemented such that on-line charging and billing is possible

Tests of the socket and socket server program showed that the socket connection can be established successfully, and information can be downloaded correctly User access can be displayed immediately on the monitor screen

5.5 System Design and Implementation

5 5 1 Charging System Model

The overall charging and billing system is illustrated in figure 5 3 The origin of the information used for charging is shown in the block titled "Network Resources & Connection Attributes" Measurement on these is made in various ways, with the resultant outputs being presented on a bank of meters The output of these meters may go directly to form a charging parameter However, in some cases meter outputs may need to be combined in order to produce the charging parameter The operation of combining meter outputs is shown as the meter process block The charging parameters is the complete set of parameters used in the charging process To denote the relationship between the meter outputs and the charging parameters, a function P representing the metering process is used

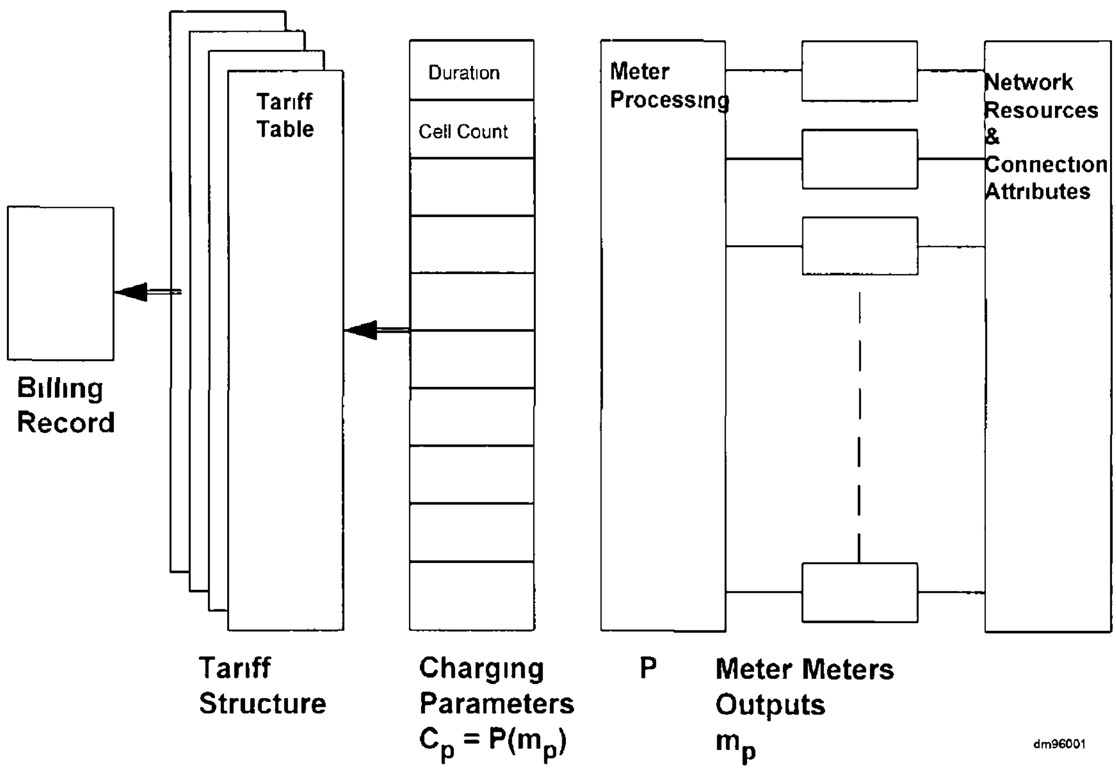


Fig 5 3 Charging Scheme Structure

As was mentioned, the charging parameter list represents the complete set of all charging parameters Possible charging parameters are considered in previous section and listed in Appendix E Some network operators may choose only a limited combination of these parameters This limited set of charging parameters will then be used to produce the price of each instant of communication A particular measured set of charging parameter values will map into one of the tariff tables The tariff table will then be used to determine the price Given that the tariff tables together determine the price of any communications instants, this set will be referred to as the tariff structure An example of what might constitute a tariff table is shown in figure 5 4

Tafiff Table X	**20-30 Km distance**	
Duration	**0-3 mins**	**10p**
	Per 1 min after	**2p**
Cell Count	**<10,000 cells**	**0 004p/cell**
	>10,000 cells	**0 002p/cell**
Further Parameters		

dm96001

Fig 5 4 Example Tariff Table

A question of particular importance, even at this stage is that of standardisation The various elements of the above process might come under this heading, for example, the charging parameters provide an interface between the customer and the network operator with regard to charging policy It may therefore be appropriate to define these parameters, and the ranges upon which they can be based A second important feature is that of measurement Parameters which are measured with only a low degree of accuracy would not be acceptable to users Therefore either the measurement technique or the degree of accuracy assumed in the measurement might need to be specified These are items which need to be considered

An implementation model as described in figure 5 5 is being considered The purpose of this model is to evaluate charging schemes at both the network level and the service level At the network level bearer services such as ABR, VBR, UBR and CBR are of interest This evaluation is done in field trial through the use of a NORTEL Passport switch placed in the trial network The Passport switch will provide output to the Network Management Accounting system and subsequently produce a bill

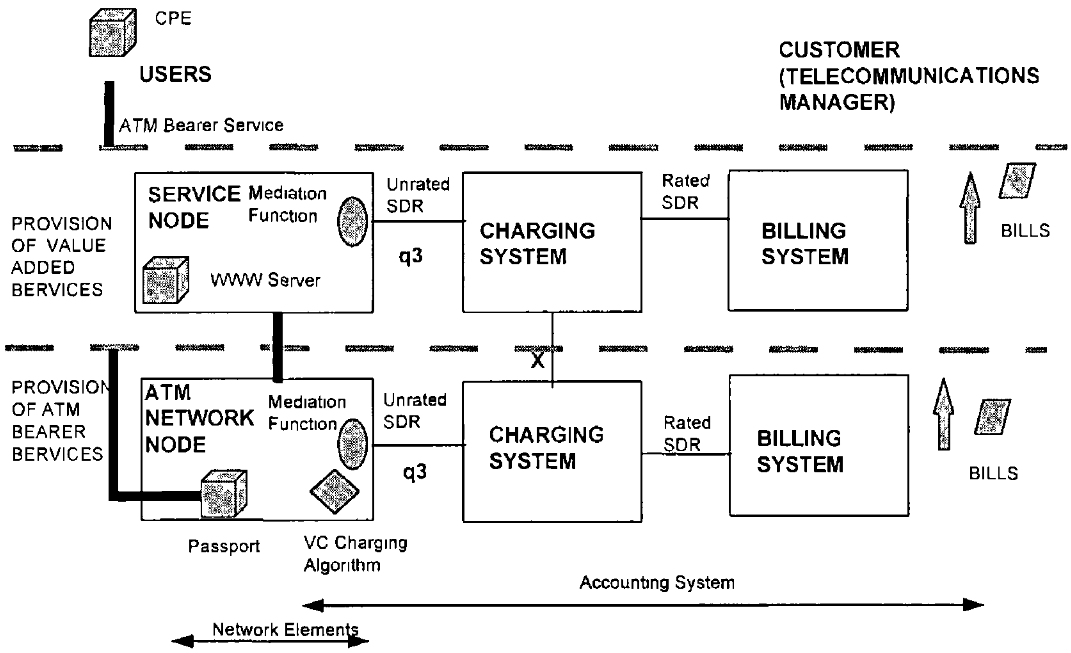


Fig 5 5 Initial Implement Proposal

At the service level, a service machine providing traffic into the network is also evaluated where the service provider is interested in billing for the service Considering how this structure should be developed in terms of providing a bill for ATM will be the focus for evaluation

The architecture supports (on a high level of definition) the accounting process through a hierarchy of 2 layers where value is added at each layer The following observations are made

(i) each layer, Network Provider, Service Provider has services
(ii) the nature of these services differs

(iii) the basic functions for each provider are the same in an accounting sense while the detailed specification differs - due to the fact that the services delivered and business requirements are different

The architecture will lead to the implementation of a prototype that can be used on more than one layer which will provide transparency between the provisioning layers The issue on how these two layers interact will be examined as the diagram above does not address the completeness of these points that an inter-layer accounting information transfer will introduce

In Fig 5 6, the system is further decomposed to reveal more details of the system design using the structure shown in Fig 5 3

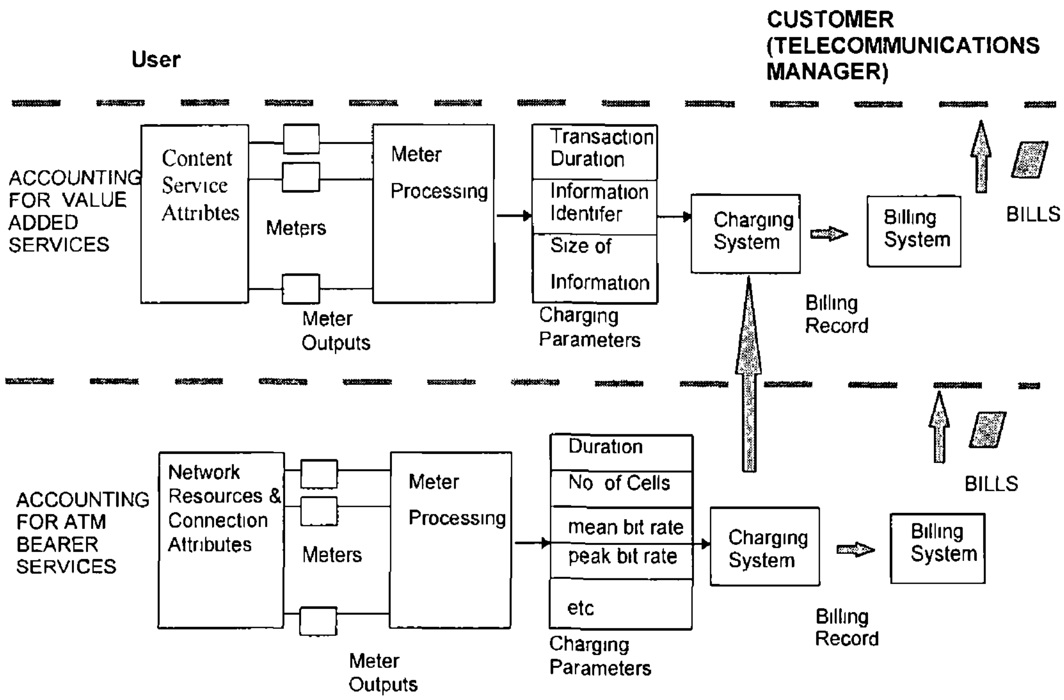


Fig 5 6 Charging and Billing System

5 5 2 ISO/IEC DIS 10164-10 2 Model

The FIONN trial charging and billing system also references to the ISO/IEC DIS 10164-10 2 draft model, which is shown in Fig 5 7

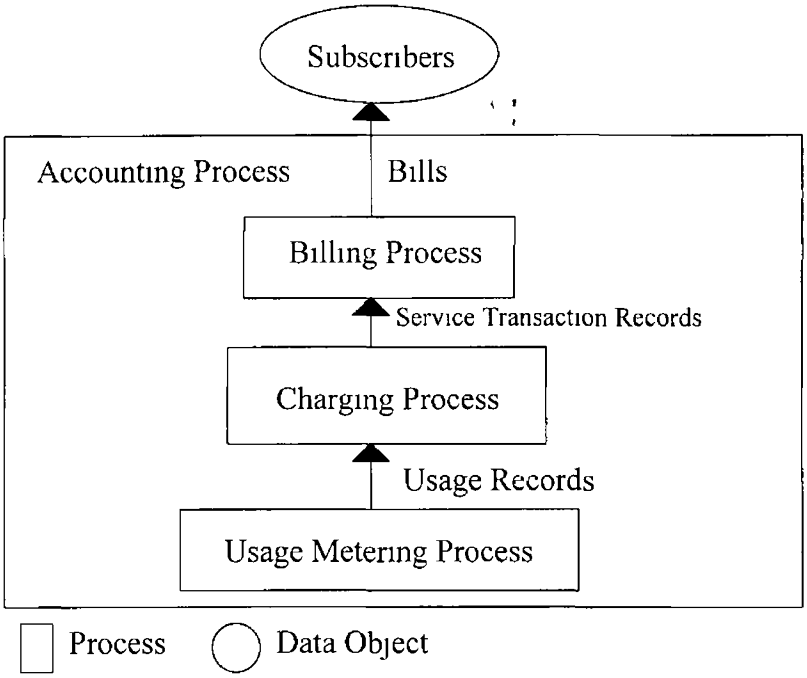


Fig 5 7 ISO Accounting Process Model

According to the ISO draft,

- the **usage metering process** is responsible for the creation of accounting records as a consequence of the occurrence of accountable events in systems
- the **charging process** is responsible for collecting the accounting records which pertain to a particular service-call in order to combine them into *service transaction records* In addition, pricing information is added to the service transaction records The process is also responsible for logging the service transaction records
- the **billing process** is responsible for collecting the service transaction records and selecting from these the ones which pertain to a particular service subscriber over a particular time-period and produce the bill from these
- the **accounting process** is also responsible for logging of the accounting records [5]

In the ISO model, *usage metering process* produces *usage records* which can be further analysed by the *charging process* and the *billing process* to present the user the bills As we shall see in the next section, a further break down of the *usage metering process* can be introduced which will lead to the mapping of our WWW service charging model to the ISO recommended model

It ıs worth notıcıng that ın the ISO recommendatıon, the usage of the words *object* and *data* looks quıte confusıng and most of the tıme ınterchangeable, e g , the word *accountable object* refers to the object whıch contrıbutes to the *usage meterıng data* In thıs thesıs, we'll assume that these two words can be used ınterchangeably, and both refer to the word data

5 5 3 Mappıng of WWW Servıce Chargıng Model to ISO Model

The reason that we want to use an ISO recommendatıon model as our chargıng system model ıs that the chargıng schemes and chargıng and bıllıng system can be standardısed ın the future

A mappıng of our WWW servıce chargıng model can be shown graphıcally

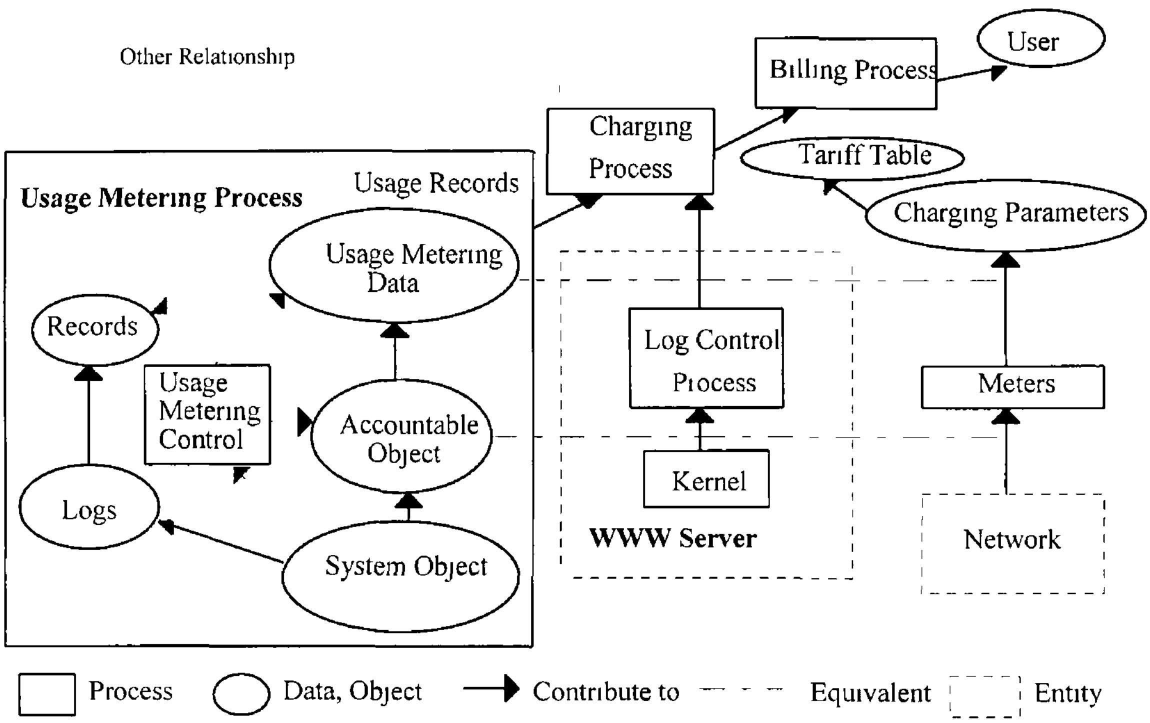


Fıg 5 8 Mappıng of ISO Model to WWW Server Chargıng Model

Accordıng to ISO/IEC DIS 10164-10 2 document, *system object* ıs all the ınformatıon contaıned ın the system, and these ınformatıon may or may not relate to the chargıng process

These system ınformatıon contrıbutes to *accountable object* whıch contaıns all the ınformatıon that maybe used for chargıng and ıs further extracted to form *usage* *metering data*, which is recorded in *usage records* and used for charging purposes The *usage metering data* or the *usage records* are then passed to the *charging process* and information produced by this process is then fed into *billing process* where bills are produced and presented to the subscribers The process in which *system object* is extracted to produce *usage metering data* and *usage records* is called the *usage metering process* according to the ISO recommendation

In our WWW server model, all the information that is contained in the server may be regarded as the *system object* The information that is available after a transaction is finished would contain certain information related to charging, i e time stamps, file length, and some other information that is non-relevant to charging purpose, like protocol type, and transaction status These can be regarded in the ISO model as the *accountable object* as they are extracted from the *system object* This information is then further processed in the *log_transaction* function to produce only the information that is related to charging purposes, which correspond to *usage metering* data and *usage* records in the ISO recommendation These *usage records* can then be further processed in the *charging process* and *billing process* to present the bill to the user

In the proposed charging model, the inputs to the meters would be all the information that contributes to charging information, and matches the description of *accountable object* in the ISO recommendation, and the meter outputs contains all the information related to charging, and contributes to the charging parameters This matches the description of *usage metering data* and/or *usage records* The meter outputs would then further be processed to produce a group of charging parameters and combined with the tariff tables already available to the system, a bill is presented to the user This could be explained in the ISO recommendation terms the *charging* and *billing processes*, where the charging parameters would be the output of the *charging process*

Chapter 6. Investigation of Service Level Charging Schemes

6.1 Introduction to Fionn Trial

The investigation of service level charging schemes is done by means of field trial The trial itself is carried out among a group of companies and colleges who are involved in a CANCAN (Contract Negotiation and Charging for ATM Networks) project, part of the European Union's ACTS program, as the resources required by the trial can not be completely provided by the college Major partners involved in this trial are Teltec DCU, Telia and Nortel

The Fionn Trial examined the charging of application services carried over an ATM Network An Internet World Wide Web server supporting a Chinese Language magazine was set up for the trial, and a public community of users were informed of its existence

Between 10 and 20 users were organised to access this magazine, and were divided into two groups About half of them were grouped into a hypothetical company named Wong & Co , while the other half remained as individual Internet users

Over 50 people were invited to dial into the home page The estimated number of people who visited is about 30 Telia made over 300 calls, and Teltec over 200, which provided sufficient data for the Charging and Billing System

6 2 Trial Description

Fig 6 1 shows the final implementation of the FIONN trial with details about configuration data The considered trial object, the WWW server, is connected to the Internet via the PASSPORT switch, the Internet gateway and the Telia Research internal data network TRABnet

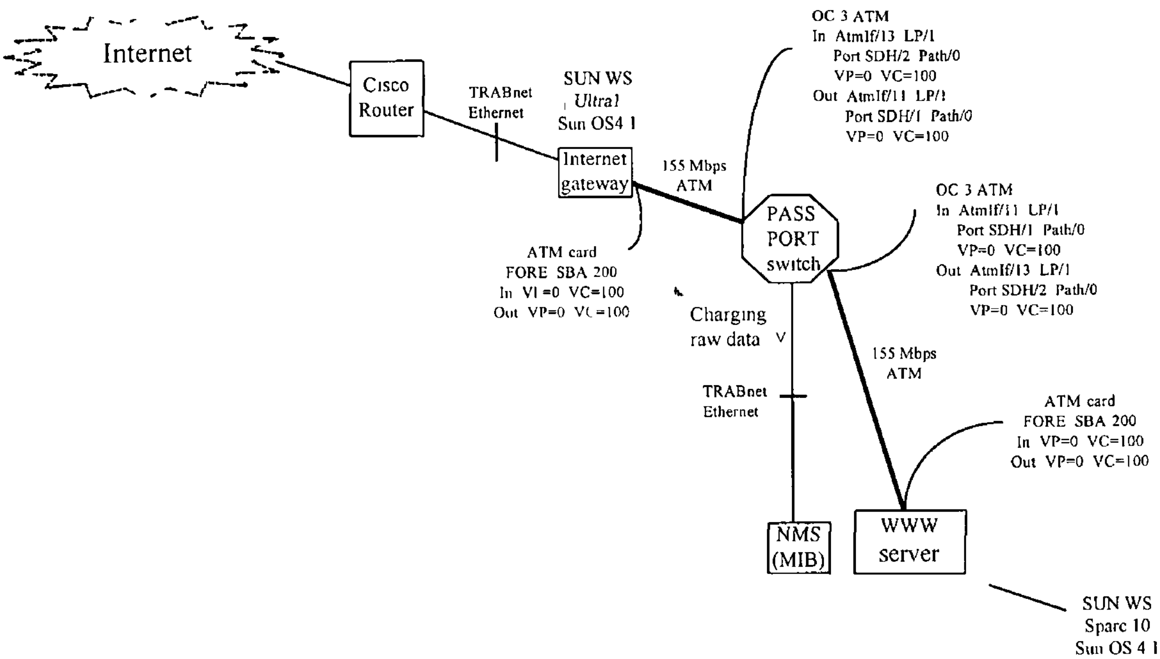


Fig 6 1 FIONN implementation in Telia's ATM lab

The Passport switch is used mainly as a meter to measure traffic statistics, with the switching functionality disabled (the software is not provided with the switch) The NMS (Network Management System) and the WWW server are installed on the same machine The NMS generates two sets of traffic log files ccLog and tpLog files, and the WWW server generates one log file

Some notable remarks

- The World Wide Web server and the NMS are hosted by the same computer
- The ATM traffic between the Internet gateway and the WWW server is connected via the PASSPORT just as loop-backs in each port, i e not connected through the switch
- All traffic to/from the WWW server is routed to the ATM connections The traffic from PASSPORT to NMS, e g charging raw data, is routed via the TRABnet
- Charging raw data recorded by the PASSPORT may contain accesses other than just to the home page of the WWW-server, due to the fact that all Telia external accesses are routed via the ATM-connections

It is assumed that the users have Internet access, IP over ATM networks are readily available, commercial services over Internet are readily available, and authentication technology is implemented

The supplier of the service is a value-added service provider, and is concerned with the provision of content via the Internet and part of the infrastructure required to deliver this content Charges levied by the ATM network operator and for Internet interconnect (on the VASP side) are the responsibility of the supplier, these charges are bundled with content and other charges for the purposes of charging the customer

The Customer is an individual Internet user Any set-up or ongoing charges levied by Internet access providers, their local network operator or others are the separate responsibility of the customer and do not directly impact on the trial or associated tariffing

6.3 Service Description

The service provided in the FIONN trial can be partitioned in two layers
From the market point of view, Fionn trial provides a web access to a Chinese magazine From the network point of view, the trial is a file transfer process

The contract between the supplier and the customer contains clauses relating to all parts of this trial description (i e service description, tariffing, system capabilities, etc) An important contract component is the definition of a *session*, i e the basis on which service usage is charged In this case, we define a session as

> *a single file transfer from the server to the user's IP address, resulting in an entry in the server's log file*

The reason to define a session as it is is to make billing easier, since a single file transfer corresponds to an entry in the log file An alternative is to define a session as one article transfer, which may involve multiple file transfer from the server to the user But this may cause difficulties distinguishing separate sessions, as the server can only record one entry for each file transfer

There will then be a charge levied for each session, as detailed in section eight, Fionn bill

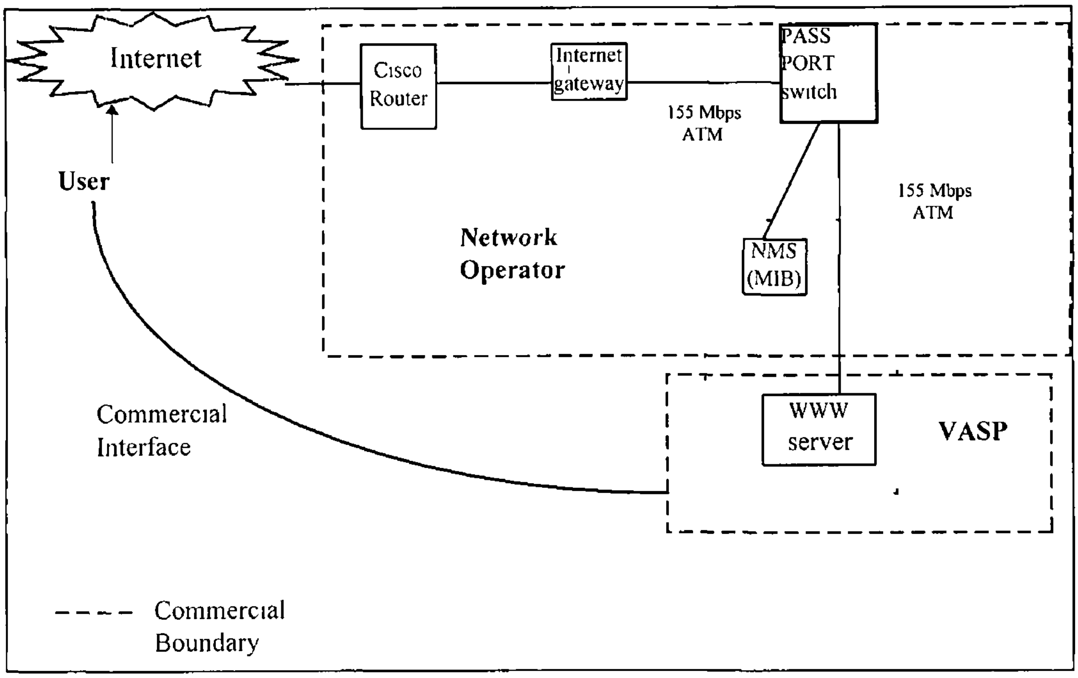


Fig 6 2 The Commercial Interface for the Fionn Trial

In a commercial environment, the contract between the supplier and the customer would be set up formally, probably on-line by means of a suitable electronic form interface Users would then be given some means of authenticating themselves for future transactions This can be implemented using the NCSA server's user authentication feature As we've mentioned in chapter 5, this feature will prompt the user for his/her user name and password Once these two pieces of information are entered by the user, the server will then have knowledge of the user's identity Since the user does not have to enter the same password and user name again to enter the same protected area as long as the user does not start a new browsing session, e g starting a new Netscape screen This feature can be used to identify individual users The password and the user name are all stored in string variables in the server program, and can be accessed by other functions

In the trial however, the above on-line contracting method is not implemented and we concentrated on charging and billing The user is assumed to have accepted the charging schemes already The only means of identifying this user in this particular trial for later transactions is his/her IP address

The commercial interface between the supplier and customer is shown in Fig 6 2 This is somewhat independent of the physical interfaces

6.4 Service Characteristics

The details about the FIONN trial are given in Appendix C This includes each produced "Intermediate"-files (cci out), corresponding start and stop date

Fig 6 3 shows the traffic characteristics for a set of Fionn recordings done by the NMS The charts are based on the Peak Cell Rate values from the *ccLOG* -files It should be noted that the diagrams are just to show the characteristics of the traffic load, without giving exact values of the Peak cell rates Further information on traffic characteristics is given in Appendix C

The "IN" curve shows the traffic into the WWW-server, and the "OUT" curve shows the traffic out from the WWW-server The vertical marks on the x-axis indicate 15-minute intervals

Note that the OUT-traffic shows a "constant" flow of approximately 200 cells/s The source of this "ghost" flow has not been localised yet, but it is probably not caused by any user

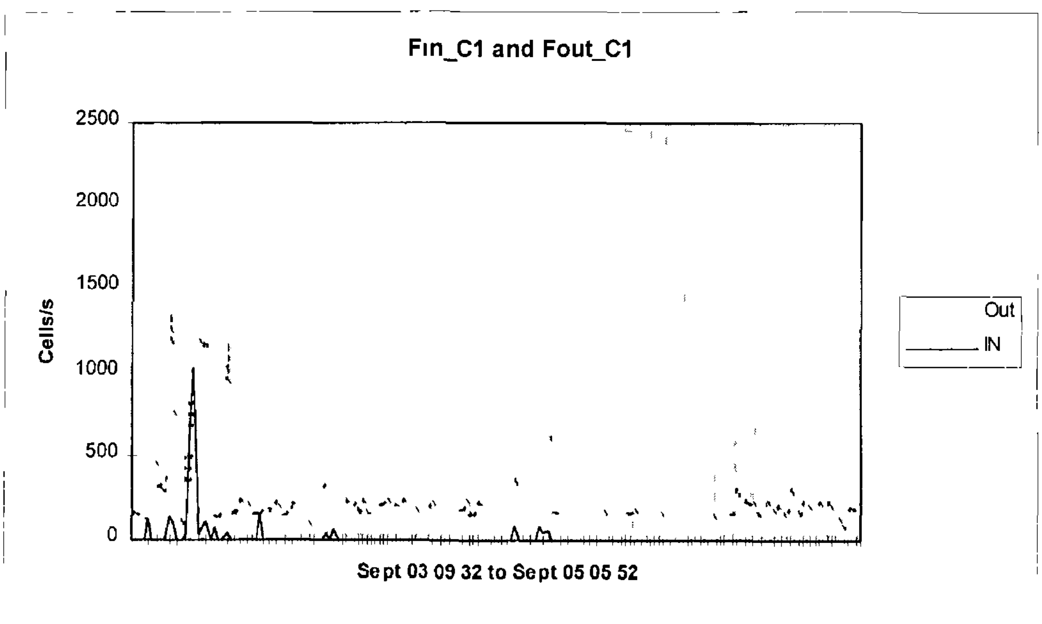


Fig 6 3 Representative Fionn recording done by the NMS

6.5 System Capability

FIONN trial used some CANCAN-specific equipment within an existing network, from which ATM traffic has been made available for our experiments

An overview of the CANCAN field trial network is given in Fig 6 4, showing the configuration of the existing ATM network and the additional CANCAN elements For the FIONN trial the switch is used to connect an Internet gateway to the WWW server

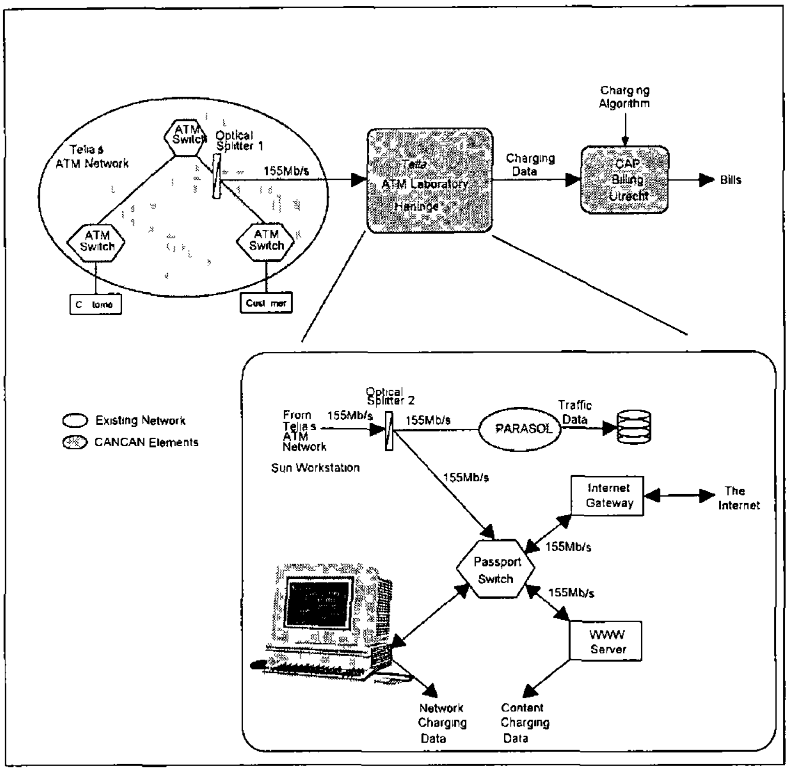


Fig 6 4 CANCAN Trial Network

The ATM network consists of two Sun Workstations, and a Passport ATM switch connected by optical links The gateway machine is a Sun Ultra running Solaris 5 using static routing with an Efficient ATM card, while the WWW Server machine is a Sun Sparc station running SunOS 4 1 3, with a FORE card The main purpose for using the SunOS system is because the WWW server software does not compile under Solaris systems, and according to the manual, one of the functions used should be strictly under 4 3BSD UNIX systems or SunOS even though the manual also claims that this function is supported in Solaris systems The gateway machine is also configured such that it doesn't have a DNS server, and hence all the IP addresses are recorded as IP numbers only It is configured in such a way, because the FORE ATM card does not work with DNS servers, and static routing had to be used for the same purpose

The server program is a modified NCSA 1 4 2 WWW Server The installation procedure, listed on NCSA's tutorial home page is very simple The server software was compiled under SunOS system

Assuming Internet with IP over ATM is readily available, as is WWW browsing software with terminal cache capability, the trial shows that cache plays a very interesting part

If a session set-up fee is included, the charging and billing system will have to be able to distinguish a real file transaction from a mere user request such as occurs when Netscape is used to call a page already held in cache, since Netscape can load pages stored in cache and leave zero amount of information transferred on the server's log Another type of log is the amount of information downloaded could be recorded as a "-" This is possibly caused by user termination of a transaction, i e during the transaction, the user hits the stop button on the browser, and forces the session to come to an end

For the purpose of the FIONN trial, and due to the nature of the service engine, only two sets of data are available

The first set of data available is directly from the service engine's log file It includes the

- Duration of transaction accurate to 1 millisecond,
- Amount of information transferred in bytes,
- Information identifier (file name)

An IP address is also recorded in the server log file, but is not sufficient to distinguish individual users Authentication technology will be implemented at a later stage of the trial, and new ID will be recorded to provide identification methods for users, instead of IP addresses For this trial, IP addresses were used as user identification

The second set of data is from the ATM switch measurement and this should provide us with information from VCs and individual cells with time stamps As this trial is IP over ATM, this information will possibly not prove to be very useful to the service provider, though it will definitely be of importance to the network provider

This data is recorded in tpLog and ccLog files Each entry in tpLog has two components

- a time stamp,
- and the number of cells received at the recorded time

CcLog files contain more ATM network related information, such as VCI and VPI

6.6 Charging Scheme

The service level and network level charging schemes we chose to use in Fionn trial is limited by the number of charging parameters that can be measured by the system It is not possible to measure statistics such as peak cell rate or cell loss ratio because the sampling interval is too long (15 minutes compared to typical web transfer which takes only a few seconds or even less), and therefore, impossible to apply charging schemes that take bandwidth and QoS as parameters (and we shall see that the cell count at the service level and the network level are different, and hence the mean cell rate)

Therefore, the network level charging scheme chosen is based on the three-term charging scheme, with the following equation

$$C = D + E\,t + F\,v$$

where the charge C is composed of

1 A constant term D representing a leasing charge, explained below
2 A duration dependent term (E t), where E is the charge per unit time and t is the duration or lifetime of the connection, and,
3 A volume dependent term (F v) where F is the charge per cell, and v is the total number of cells passed by the connection

The scenarios are summarised in Table 6 1 below and have a number of combinations of fixed charge (FC), volume based (VB) and duration based (DB) coefficients (D,E,F in the three term formula) The term D represents a leasing charge of £26,610 per month

Scenario's	CS	coefficient D	coefficient E	coefficient F
1	fixed charge	used	0	0
2 1a	volume based	0	0	used
2 1b	volume based	0	0	used
2 3	volume based	0	0	used
3	duration based	0	used	0
4 1	FC+VB	used	0	used
4 2	FC+VB	used	0	used
4 3	FC+VB	used	0	used

Table 6 1 Charging Schemes for the Fionn Trials

The first scenario (1) used only fixed charging (FC)

The next scenarios (2 1a, 2 1b and 2 3) used a purely volume-based approach

One scenario (3) used a duration-based approach However, for the Fionn trial, cells are counted every 15 minutes If the counters are not synchronised to a start of a transmission, then calls of a small duration (e g Web Page transfers of only a few tens of milliseconds) could be missed entirely

Three scenarios (4 1, 4 2 and 4 3) combined fixed and volume-based charging the fixed charge must be competitive However, it can be used to penalise short term use of a network, where both VB and FC are combined in one charging scheme

In the trial, the duration term is dropped, since most web transfers fall into the category of ABR, and by the analysis given in chapter 3, duration term is not used Also dropped is the subscription term, which does not have a significant influence on the analysis if it is applied monthly or quarterly However, it will become significant if it is applied each time when a connection is set-up, but since the switch used in the trial can not tell one connection from another, the effects are not analysed

6 7 Price Plan

The price plan includes three parts, similar to the three term charging method

$$TotalCharge = ContentCharge + E' \times Duration + F' \times Volume$$

where E' and F' are constants, and ContentCharge is a constant charge determined by the content of the file required by user It is analogous to the network level charging scheme, only the fixed rate subscription fee is changed to content charge

Even though the network scheme and the service level scheme look surprisingly similar, they are developed independently The network level scheme is limited by the nature of the hardware, and the service level scheme is limited by the web server, which is only capable of recording the parameters listed in section five

The pricing plan is given in the following table, with currency units in US$

Proposals	Duration	Size	Information Indicator	Remarks
A			Index files(0) free Articles(2) 0 15 US$ Big Images(1) 0 015 US$	Estimated value of a single article in a typical magazine
B		0 015 US$ per 10,000 bytes		PSTN, international rate, 75 cents per minute
C	0 015 US$ per second			Arbitrary value
D		0 015 US$ per 10,000 bytes	Index files(0) free Articles (2)0 15 US$ Big Images(1) 0 015 US$	

Table 6 2 Tariffing for the Fionn trial

6.8 FIONN Bill

An example Fionn Bill is shown in Table 6 3

Only usage based charging is considered in this bill The final bill produced is shown below

Items	Duration (Sec)	Volume (Cell #)	Inf ID	Charge A (cents)	Charge B (cents)	Charge C (cents)	Charge D (cents)
1	0 065	76	2	15	0 540	0 098	0 54
2	0 017	6	0	0	0 038	0 026	0 038
3	0 054	489	0	0	3 510	0 081	3 51
4	0 030	3	0	0	0 017	0 050	0 017
5	0 023	139	0	0	1 010	0 035	1 01
6	0 047	512	0	0	3 690	0 071	3 69
7	0 023	7	0	0	0 044	0 035	0 044
8	0 050	515	0	0	3 760	0 080	3 76
9	0 056	368	0	0	2 640	0 084	2 64
10	0 023	8	0	0	0 042	0 035	0 042
11	0 027	293	0	0	2 100	0 041	2 10
12	0 414	925	0	0	6 660	0 620	6 66
13	0 041	403	0	0	2 900	0 062	2 90
14	0 019	6	0	0	0 041	0 029	0 041
15	0 023	9	0	0	0 060	0 035	0 060
16	0 021	382	0	0	2 750	0 032	2 75
Total		4141		15	30	1 5	45

Table 6 3 Example of a FIONN Service Level Bill

The bill for the bearer service for the above traffic is also shown here for comparative purposes The subscription fee and the constant for duration charging E are both set to zero

$$Charge = Volume \times K(\$ \ per \ Mbps)$$

This charging scheme is designed for UBR traffic For method 1, K is chosen as 0 001, and K is 0 0001 for the second method this would be the range of possible choices for K

Duration	Cell Count	Charge I (cents)	Charge II (cents)
14 03 54 10 - 14 04 01 97	4849	0 19	0 019

Table 6 4 Example of FIONN Network Level Bill

From the two bills, we can find some very interesting things

First, the cell count for the service level bill and the network level bill is different A careful examination of the switch log files shows that there is a small amount of traffic going from the gateway machine to the machine where the WWW server resides even when there is no user request The increase is not very significant, but can be clearly seen The nature of this traffic is not yet clearly known However, the provider of the PASSPORT switch, Nortel, claims that the data collection software knows it is functioning by detecting an increase in cell count in its log file This suggests that there might be a mechanism implemented in the switch such that every few seconds, the switch generates a small traffic so that the software can see the increase in cell count As to the increase in the cell count, a large proportion is due to the overhead TCP/IP layers put on the ATM cells

A calculation is done to find approximately how many extra cells are created due to the TCP/IP layers We assume that each IP packet carries 64 bytes of information, and that it does not contain the option field Without the headers that the TCP protocol adds on to the information, the extra number of cells amounts to about 200-300 We can estimate then, that amongst these about 700 extra cells, about 400-500 are due to the creation of the TCP and IP headers

A further analysis of the total amount of traffic in the trial shows some astonishing figures

Hits counted by the server 2325

Total amount of duration recorded by the server 2901 74 seconds

Total cell transferred by the server ~550,000

compared to the switch statistics

Total duration 16 days

Total amount of cell registered by switch ~3,600,000

We notice that the difference between network registered traffic and server registered traffic is huge (about 7 times the amount registered by the server) However, if we divide the extra traffic by the 16 day duration, it only results in about 2 extra cells per second, which is just as what we have suspected the switch generated traffic just to keep the Nortel data collection software alive

However, this raises another question, which is, who should pay for this extra amount of traffic? Should it be the network provider, or the service provider, or even the web subscribers? We think that most web surfers will not accept a bill for extra TCP/IP information What people see on the web is meaningful information, either in text, or graphic, or audible information It doesn't make any sense to tell the user to pay for TCP or IP headers It should be, we think, the service provider who pays for the bill, since they are the ones who are selling the traffic, and profit from it And it is this exact traffic that causes all the extra bits So the network provider should charge the SP for this extra traffic

Another interesting point is that we can see a huge amount of profit comparing the NP's and SP's bills Examining table 6 3, we can see that scheme D produces a way too high bill for the users (charging by content and duration together), which means that the figures we have chosen might be too high, and the service level charges should be lowered

6.9 Evaluation

The Fixed charging scheme used for the Fionn Trial was evaluated using a set of criteria used in CANCAN A detailed description and full list of the criteria is given in Appendix B The table as shown below is given for only the fixed rate scheme with marks from 1 to 10 for each criteria The highest mark is 10 A full explanation of the analysis used is given in Appendix A

No	Criteria	Score
1	Usage Sensitivity	0
2	Practicality	9
3	Fairness to User and Network Operator	5
4	Intelligence Required of User or Applications	10
5	Predictability	10
6	Generality	10
7	Expandability	0
8	Robustness	10
9	Dynamic	0
10	How easy to fool?	10
11	Profitable to User and Operator	5
12	Co-operative Sharing	0
13	Auditability	10
14	Contract	5

Table 6 5 Evaluation for fixed rate scheme

During the trial, several problems were encountered, some of them solved without any difficulty, while some remain unsolved throughout the trial For example, the machine where the WWW server resides remained unstable during the field trial, and most of the users trying to connect to the test site experienced difficulties getting into the web page This is viewed by the user as a poor quality of service, and should be reflected in the charging scheme However, this is due to the network instability, and should be compensated by the network operator This means that there should be a correlation between user's definition of QoS and network operator's QoS definition Another one would be the ghost traffic, which raised the question who should pay for this traffic

Chapter 7. Conclusions

7.1 Conclusions

This project investigated some of the charging schemes, both at the service level and the network level In this section, we will discuss the different features of these charging schemes, and their feasibility

The network level charging scheme we proposed in chapter 4 is an Internet/LAN charging scheme It can be used by colleges and/or companies to distribute their Internet bill to individuals or departments The scheme we proposed here is based on volume charging method, and the only parameter that it takes is volume The simplicity means that applying the charging scheme does not require additional hardware (a router capable of measuring traffic statistics and storing them in internal memory can provide enough charging information) In the trial, we used a CISCO 2514 router and CISCO's own MIB The reason that we've chosen a proprietary MIB rather than a standardised MIB is because that CISCO router only supports two standard MIBs SNMP and RMON Neither of the two MIBs provide enough charging information Feature of RMON II MIB was not thoroughly investigated, since no RMON II software was available from CISCO by the time of purchase However, we expect the required charging information will be supported by RMON II or future releases of RMON

The charging method used by the external line network provider was assumed to be fixed rate charging in the field trial, but in reality, this can be any ATM charging scheme, e g three term charging, or usage based charging Our proposed charging method does not depend on the way the NP charges the company/college, but is only concerned with how to distribute the NP's charge to each department fairly Using volume based charging method, we ensured that charges produced by our scheme is directly proportional to each department's or user's usage And from a network usage point of view, this scheme is quite fair to everyone

One additional feature of our network level charging scheme is that it can give user indication of his end of month bill, and the user can reduce his network usage if he decides he does not want to spend too much

Another important factor is that the field trial also showed the prediction method is quite good for both bursty and non-bursty network traffic, which means that it should work well regardless of the type of network traffic

Percentage of weight is not given in this thesis, though arbitrary values were chosen in the field trial The reason being that the choice depends on traffic pattern of each department/user When implementing the charging scheme, the charging and billing system should allow the system administrator to vary the percentages to find the best value for a particular group of users

We expect that this charging scheme be easily accepted by users, since

- it is very simple, which means it is easy to implement, and the cost of implementation is low,
- it gives prediction on end of month cost, which can help user control his monthly expense

The service level charging scheme proposed in this thesis is again, very simple, based on only duration, volume and information As we have explained in chapter 3, existing Internet service charging scheme is fixed rate charging scheme (rent for network access) which does not reflect the users' need (quality of service, content of service), or no charge at all (service level, web pages) which doesn't make any profit for the content provider We would expect this existing scheme will change in a very short period, to put more focus on QoS, information content and traffic statistics Our charging scheme is designed for this need The reason that our charging scheme did not take into account of traffic statistics is that the SP won't need to introduce extra hardware if our charging scheme is adopted As most web servers nowadays provide more information in their log files than the common format log file (e g the new NCSA server provides duration), the SP can directly change to our charging scheme without any change at all to his system The simplicity of this charging scheme is also its biggest advantage

Despite its simplicity, this charging scheme also provides great flexibility As we have shown in our field trial, choosing different parameters for the three term charging scheme results in completely different charging schemes, such as duration charging, volume charging, or content charging The SP can use any one of the charging schemes he wants, or any combination of the schemes And another advantage of this charging scheme is its predictability Apart from duration charging, both volume charging and content charging method can be predicted by the user, i e the user can easily estimate how much he is going to pay for the information he wants to access if either volume or content charging is used Thus, the user will always know how much he spent before he gets the bill, and this feature can make this scheme more popular

In the field trial, our parameter values are chosen according to telephone rates and estimated value of an article in a typical magazine We can expect the users willing to accept the charging scheme, as the charge is not higher than the price of buying a newspaper On the other hand, we also showed in the trial that the profit margin of the SP compared to the NP is very big This means that competition will definitely bring down the charge, which makes web browsing even cheaper than buying a newspaper or magazine Another advantage to the user is, he does not have to buy a magazine or newspaper, just to read one article Browsing on a web server using our charging scheme does not only allow the user to pay less, but also allows him to read what he wants to read, without paying for the rest of the information that he is not interested in This feature of our charging scheme will definitely make web browsing more attractive

However, if a session set-up fee is applied in the charging scheme, the user may be charged for information that he didn't download For example, Netscape has the capability of examining the user requested file and files stored in its cache, and download from cache instead of the remote server But the remote server will see a user request, and put this in the log file If a set-up fee is applied, the user will be charged for a set-up fee while he actually did not download the information from the server

A combination of volume charging and content charging or simply content charging is recommended if a service provider is going to use our scheme Web service is different from telephony services, and the longer time it takes for a user to access a file, the worse the quality of service is This will make duration charging hard for users to accept On the other hand, users are always willing to pay more for more useful information (Larger volume can sometimes imply more information, and hence, may also be used)

We also recommend a set of parameters being used as CDR (Call Data Records) They are

- Duration
- Volume
- Information indicator

we would like to keep duration as a parameter in the CDR, as a measurement of quality of service, and the SP can also use it in the charging scheme e g providing discount to user if duration is too long But duration charging is not recommended

7.2 Design Recommendations for Internal Billing Software

7 2 1 Communication with SNMP Agent

There are several ways of retrieving MIB data from SNMP agent in the router to a database residing in a PC The one that is used in the trial program is the easiest, in terms of programming, as it involves minimum amount of coding and is very easy to understand However, it does require external software package, including two parts SNMP++ from HP, which does not need a license, and can be used free of charge, SNMP dynamic link library, which is supplied by two companies, FTP Software and AceCom as a commercial product

There are however, two other ways of doing this One is to use a direct link between the serial port on the PC to the console port on the router This will require setting up of a serial line communication, and the set-up of the serial port on the PC depends on the type of the router The user can then use his program to issue commands that retrieve charging related information, which again, depends heavily on the version of the software running in the router Any change at all in the model of the router/switch, and any upgrade of the router software, might require a corresponding modification in the software, which is not a good design method But it does, however, avoid using commercial software products, and hence, eliminates possible license agreement problems

Another approach is to use an SDK product provided by MG-Soft, a Slovenian company This software is compatible with Visual C++ 4 0, Borland C++ 5 0 and Delphi 2 0 It also supports SNMP++ from HP But when trying to design the same program using Borland C++ Builder Professional, a similar product as Delphi 2 0 using SNMP++, we have encountered various compilation and link time errors, and the worst thing when using C++ Builder is, the compiler is not able to recognise the format of some library files that come in binary form

I would hereby recommend programmers to use HP's SNMP++ as the software package to communicate with SNMP agent on the router, as it is easy to program (takes less than 10 lines of coding to perform a single GET command on SNMP agent, as opposed to more than 40 lines using ordinary SNMP functions in C) Another reason is that ordinary SNMP functions only allows user to retrieve SNMP data column by column, while this package SNMP++ allows user to retrieve them row by row Field test shows that retrieving 300 single SNMP items takes about 10 seconds, and a large company or university could have much more than 300 items m the SNMP MIB that relates to charging, which could in turn take much longer time If we could only retrieve information in a columnar fashion, while data related to one user is stored in a row fashion, by the time the software retrieves the second or third data related to a user, the value for the first entry might have already been changed, as the system requires a significant amount of time for data retrieval Hence, data related to a single user should be retrieved at the same time, and SNMP++ is the only software package for 32 bit Windows systems that can achieve this

7 2 2 Database Connectivity Issues

As there is already a piece of software developed using Borland Delphi 2 0, and Paradox database, it is desirable that our system can also use Paradox database engines as well This can be easily achieved using Microsoft's ODBC drivers, as ODBC driver for Paradox is available on Internet for Microsoft Office users

7 2 3 Other Issues

There are other issues that should be taken care of when making this program

1 **Byte counter value** The counter stores accumulative value of all the bytes going from source to destination The number of bits used to represent this value is not known yet, but there are two types of counters for SNMP MIBs, one uses 32 bits, and the other uses 64 bits I would expect the counter to use 64 bits, but the actual number of bits it uses should be found on Cisco's documents As the counter record accumulative values, the counter value will eventually go back to zero and restarts again If a program only reads the value in the counter, and does not take that fact into account, then it will surely give wrong bills to users

2 **Number of days in a month** In the trial program, it is assumed that the number of days in a month is 30, but this value various according to the month, and the year, so the programmer should pay attention to this fact as well, when making a commercial product, especially February

3 **NP's charging method** Again in the trial, leased line style charging is used for simplicity, but in the actual program, the programmer should allow the user to specify the type of charging method used by the network provider, in order for the program to make any predictions on end of month user expense

4 **New users** In the trial, it is assumed that every user is already present from the beginning of the month, and no new users are added during the month However, this does not have to be true, and the actual software should take care of new users coming in the middle of the month, and should be able to deal with this situation

References

1 Jinsheng Li and Minoru Akyiyama, *ISDN & ATM*, Science Publishing House, 1996

2 R J Edell, N McKeown, and P P Varaiya, *Billing Users and Pricing for TCP*, IEEE Journal on Selected Areas in Communications, Vol 13, No 7, September 1995

3 D E Eastlake 3rd, *An Application Level Internet Payment Syntax*, Internet Draft, Internet Engineering Task Force

4 R G Cole, D H Shur, C Villamizar, *IP over ATM A Framework Document*, Internet Draft, Internet Engineering Task Force

5 *Information Technology - Open Systems Interconnection - Systems Management Usage Metering Function*, ISO/IEC DIS 10164-10 2, Draft International Standard

6 Jeffrey K MacKie-Mason, Hal R Varian, *Pricing the Internet*, Public Access to the Internet, Brian Kahin and James Keller, eds Cambridge, MA MIT Press,1995 269-314

7 Karim Traore, *Introduction to Charging and Pricing in Current and ATM Networks*, Ascom Sophia Research Lab, Oct 13th 1994

8 T Berners-Lee, R Fielding, and H Frystyk, *Hypertext Transfer Protocol - HTTP/1 0*, HTTP Working Group, Internet-Draft <draft-ietf-http-v10-spec-03 ps>

9 F Halsall, *Data Communications, Computer Networks and Open Systems*, Addison-Wesley 1992

10 M Laubach, RFC 1577, *Classical IP and ARP over ATM*, Internet Engineering Task Force, Network working Group, Internet Request for Comments

11 Zahir Ebrahim, *A Brief Tutorial on ATM*, http //juggler lanl gov/lanp/atm tutorial html

12 F P Kelly, *Tariffs Policing and Admission Control for Multiservice Networks*, 10th UK Teletraffic Symposium, April 1993

13 F P Kelly, *Charging and Accounting for Bursty Connections*, Internet Economics, MIT Press, 1997 253-278

14 C Courcoubetis, G Kesidis, A Riddei, J Walrand and R Weber, *Admission Control and Routing in ATM Networks using Inferences from Measured Buffer Occupancy*, IEEE Trans Comm , Vol 43, Nos 2/3/4, pp1778-1784, April 1995

15 Biørn Vierø, *Traffic Measurements on Variable Bit Rate (VBR) Sources with Applications to Charging Principles*, Computer Networks and ISDN Systems 20, Elsevier Science Publishers B V (North Holland), 1990, pp435-445

16 S Low and P Varaiya, *An Algorithm for Optimal Service Provisioning using Resource Pricing*, IEEE/ACM Transactions on Networking 1(5), IEEE, Oct 1993, pp547-553

17 F P Kelly, *Effective Bandwidths at Multi-Class Queues*, Queuing Systems 9, 1991, pp5-16

18 R J Gibbens and P J Hunt, *Effective Bandwidths for the Multi-Type UAS Channel*, Queuing Systems V 9, 1991, pp17-28

19 D E Eastlake 3rd, R Khare and J Miller, *Selecting Payment Mechanisms over HTTP*, Joint Electronic Payments Initiative, World Wide Web Consortium, Internet Draft <draft-khare-jepi-uppflow>

20 W Stallings, *SNMP SNMP v2 and RMON Practical Network Management 2nd edition*, Addison-Wesley 1996

21 A K Larsen, *All Eyes on IP Traffic*, Data Communications, Mar 21, 1997

List of Appendices

Appendix A. Evaluation of Charging Scheme Used in FIONN Trial

This evaluation is only applied to fixed priced charging

1 **Usage Sensitivity** None

2 **Practicality** Excellent

3 **Fairness to User and Network Operator** *User* Users may appreciate a fixed charging scheme (FCS) if they make considerable use of the network and its resources, if not, a fixed charging scheme is unfair to the user An FCS is predictable, but on the other hand, it does not take account of the network traffic, which is sometimes also important to the user because the traffic can affect the QoS A WWW user would not expect to pay the same price for a 10 second and a 1 hour transaction of the same HTML document ***Network Operator*** The situation is reversed for the NO The charging scheme does not itself allow the NO to predict maximum mean and peak cell rates Such a facility would be useful to assess the effect a user may have on network traffic Network operators may also like a charging scheme which gives them the possibility of tuning prices for network resources according to anticipated demand

4 **Intelligence Required of User or Applications** This may appear to be minimal However an FCS may actually require some intelligence of the billing system For example, an HTML page may consist of a main text file, with some small images, which are separate files, now form the WWW server's point of view, the transaction of a single file is a session, and the user only sees one screen full of information as one session, however if an HTML has 10 small images together with the main text file, the server will actually see and log eleven transactions when this particular page is accessed by the user The billing system will have to have some knowledge as to how to group several sessions logged by the server to a single user session

5 **Predictability** The tariff is known before a call There will be no surprises for the customer There may be surprises for the NO, unless he or she puts bandwidth limits on a user's traffic (e g by using bandwidth limiting or bandwidth shaping mechanisms) While an FCS is predictable, it does not take into account the network

traffic, this network traffic is sometimes also important to the user, because the traffic can affect the QoS A user would not expect to pay the same price for a 10 second and 1 hour transaction of the same document

6 **Generality** Excellent

7 **Expandability** None

8 **Robustness** Excellent

9 **Dynamic** The FCS does not cope on-line with changes in a user's traffic

10 **How easy is it to fool the charging scheme?** A fixed charge is always paid irrespective of the cell volume and cell rate of a user's source The amount of traffic a user sends into a network is not limited by an FCS, only by bandwidth limiting mechanisms such as UPC

11 **Profitability to User and Operator** Users may appreciate a fixed charging scheme if they make considerable use of the network and its resources, if not, an FCS is unprofitable to the user The situation is reversed for the NO

12 **Co-operative Sharing** Because of a lack of (1) a traffic limiting or prediction mechanism, and (2) a mechanism for allocating dynamically the user's use of the network resources and (3) a timing component, the charging scheme does not foster the sharing of network resources by a multiplicity of users

13 **Auditability** Excellent

14 **Contract** The FCS charging scheme take account of the contract between customer and provider However, this contract gives no information which could constrain the maximum bandwidth available to the user, and which could thus allow the NO to predict the effect the user's traffic will have on the network

Appendix B. Criteria for Comparing Charging Schemes

This Criteria List is used for an initial assessment of charging schemes

1 **Usage Sensitivity** How sensitive the CS is to the usage of network resources Does the CS take into account peak bit rate and statistical properties such as mean bit rate and burstiness? For example, if the model for charging scheme is defined for constant bit rate sources or for peak bit rate allocation of resources it can be regarded as partly usage sensitive

2 **Practicality** How easy and obvious is the implementation of the CS? For example, does the implementation of the CS depend on how it will interact with the CAC and UPC functions?

3 **Fairness to User and Network Operator** ***User*** Users may appreciate a CS if it is usage based If a mean rate or some unfamiliar characteristics have to be declared in advance some users may end up paying for more bandwidth than they actually used Alternatively, if an absolute maximum bandwidth is declared in advance, as is effectively the case with POTS, then users pay only by time duration and the CS offers "no surprises" ***Network Operator*** The network operator (NO) may also appreciate a CS offering "no surprises" since it enables them to predict more effectively the effect of a particular user on the overall link and network traffic, especially where telecommunication costs are high percentage of the NO's revenue Network operators may also like a CS which gives them the possibility of tuning prices for network resources according to demand

4 **Intelligence Required of User or Applications** How much does the user need to know about the workings of the CS? How understandable to the user is the contract negotiation process, for example, does the user need to pre-declare detailed traffic statistics, the complexity of which may not be commensurate with his mathematical

knowledge? How much experience does the user need in order to maximise his benefit when employing the particular CS?

5 **Predictability** Is the tariff known before a call? It may not be if it depends on some statistical properties of a source other than the simple mean and peak How easy is it to predict the long-term charge? This could change if the CS involves a periodic re-negotiation process with the network operator Basically, most users appear to want a CS which offers "no surprises"

6 **Generality** To which service categories may the CS be applied? As a reminder, the ATM Forum has identified the following categories CBR, rt-VBR, nrt-VBR, ABR and UBR, to which the ITU has added the ABT category See the ATM Glossary, section 23) We may eventually see a situation where different CS's - each for different service categories - coexist in an ATM network

7 **Expandability** Can the CS be expanded, extended or modified to accommodate the changing aspirations and requirements of users, customers, network operators and service providers? Can the changes be carried easily and in a way which is understandable by, fair and equitable to all interested parties?

8 **Robustness** How sensitive are the calculations performed by, and the results obtained from, the CS, to implementation factors such as rounding errors, the realisation of algorithms, the realisation of statistical procedures, the programming language used and the nature of the measurements and metering required from the ATM network? How sensitive are the results obtained by the CS to variations in the user's traffic pattern?

9 **Dynamic** To what extent, and how quickly, can the CS adapt or respond, on-line, to changes in a user's traffic?

10 **How easy is it to fool the CS?** This may be possible if the approximations or theoretical assumptions used by the algorithm, on which the CS is based, are known For example, a CS which samples bandwidth at regular intervals could be fooled by a

user if he or she sends a traffic burst in between the sampling times If the CS has complex interaction procedures with other network functions such as CAC and UPC such a situation could also potentially arise

11 **Profitability to User and Operator** If a CS is usage based, then employing it could potentially be profitable for customers and network operators alike Knowledge of the nature of, and experience in using, a particular CS could help customers and operators to maximise their respective gains when employing the CS

12 **Co-operative Sharing** How well do particular CSs foster the sharing of network resources by a multiplicity of uses? For example, dynamic charging has a built-in a mechanism for pricing according to the demand for network resources Other schemes may require the pre-declaration of parameters which can facilitate the effective and optimal management of network resources by the NO to the benefit of the users of that network

13 **Auditability** Can a user's actual use of the ATM service be traced in response to an audit query from the customer? The ease with which this can be done will depend on the extent and accuracy to which the user's traffic is monitored It will also depend on how well the procedure for calculating the charge is known, whether this is known in advance, and indeed, how easy it is to understand how it works and how it may change while a call is in progress

14 **Contract?** Does the charging scheme implicitly take account of the contract between customer and provider? This question is related to that of predictability

Appendix C. FIONN Traffic Records

The FIONN trial carried out several short network measurements.

Three types of data files has been produced.

The following table shows the details about the FIONN trials - each produced "Intermediate"-files (cci.out), corresponding start and stop date, lasting in hours, and the directory name in the CANCAN server where the trial data can be found.

Intermediate- file *cci.out*	**Start date**	**Stop date**	**Hours**	**Server directories for trial data**
Fin_A_cci.out	1996-08-30 15:59	1996-08-30 18:14	2	FIONNIN/Aug3001
Fout_A_cci.out	1996-08-30 15:57	1996-09-01 17:47	26	FIONNOUT/Aug3001
Fin_B_cci.out	1996-09-02 15:44	1996-09-02 17:29	2	FIONNIN/Sep0202
Fout_B_cci.out	1996-09-02 15:45	1996-09-02 17:16	1,5	FIONNOUT/Sep0202
Fin_C1_cci.out	1996-09-03 09:32	1996-09-05 05:52	44	FIONNIN/ Sep0305
Fout_C1_cci.out	1996-09-03 09:32	1996-09-05 05:52	44	FIONNOUT/ Sep0305
Fin_C2_cci.out	1996-09-05 06:22	1996-09-05 14:53	8,5	FIONNIN/ Sep0305
Fout_C2_cci.out	1996-09-05 06:07	1996-09-05 14:53	9	FIONNOUT/ Sep0305
Fin_D_cci.out	1996-09-06 13:43	1996-09-06 23:14	9,5	FIONNIN/ Sep0606
Fout_D_cci.out	1996-09-06 13:43	1996-09-06 23:14	9,5	FIONNOUT/ Sep0606
Fin_E_cci.out	1996-09-07 18:11	1996-09-08 10:42	16,5	FIONNIN/ Sep0708
Fout_E_cci.out	1996-09-07 18:11	1996-09-08 10:42	16,5	FIONNOUT/ Sep0708
Fin_F_cci.out	1996-09-13 14:19	1996-09-13 15:19	1	FIONNIN/ Sep1313
Fout_F_cci.out	1996-09-13 14:19	1996-09-13 15:04	1	FIONNOUT/ Sep1313
Fin_G1_cci.out	1996-09-16 10:25	1996-09-16 14:25	4	FIONNIN/ Sep1617
Fout_G1_cci.out	1996-09-16 10:25	1996-09-16 14:25	4	FIONNOUT/ Sep1617
Fin_G2_cci.out	1996-09-16 16:00	1996-09-16 19:00	3	FIONNIN/ Sep1617
Fout_G2_cci.out	1996-09-16 16:00	1996-09-16 19:00	3	FIONNOUT/ Sep1617
Fin_H_cci.out	1996-09-17 15:29	1996-09-19 21:05	53,5	FIONNIN/ Sep1720
Fout_H_cci.out	1996-09-17 15:44	1996-09-19 21:05	53	FIONNOUT/ Sep1720
Fin_I_cci.out	1996-09-20 15:18	1996-09-21 21:51	30,5	FIONNIN/ Sep2021
Fout_I_cci.out	1996-09-20 15:18	1996-09-21 21:51	30,5	FIONNOUT/ Sep2021
Fin_J_cci.out	1996-09-23 16:57	1996-09-23 22:58	6	FIONNIN/ Sep2323
Fout_J_cci.out	1996-09-23 17:00	1996-09-23 23:00	6	FIONNOUT/ Sep2323

The following figures Fig A 3 1 to Fig A 3 12 show the traffic characteristics for all FIONN recordings done by the NMS The curve charts are based on the Peak Cell Rate values from the *ccLOG* -files It should be noted that the diagrams are just to show the characteristics of the traffic load, without giving exact values of the Peak cell rates The vertical marks on the x-axis indicates 15-minute intervals

The "IN" curve shows the traffic into the WWW-server, and the "OUT" curve shows the traffic out from the WWW-server

Note that the OUT-traffic shows a "constant" flow of approximately 200 cells/s The source of this "ghost" flow has not been localised yet, but it is probably not caused by any user

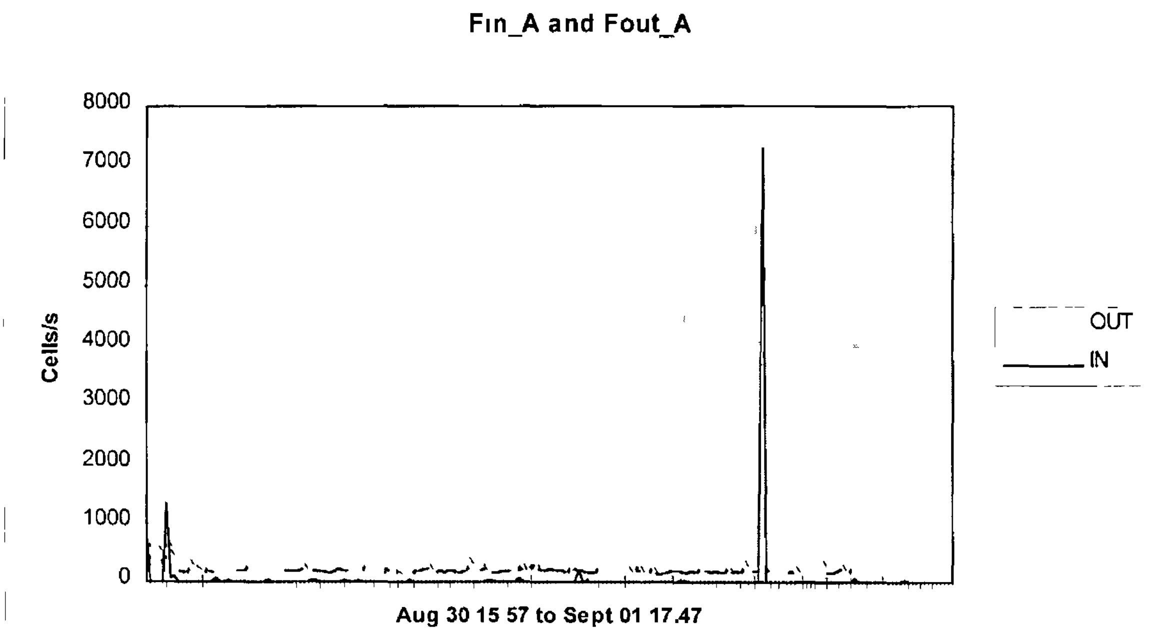


Fig A 3 1 FIONN recordings by NMS during Aug 30 and Sept 01

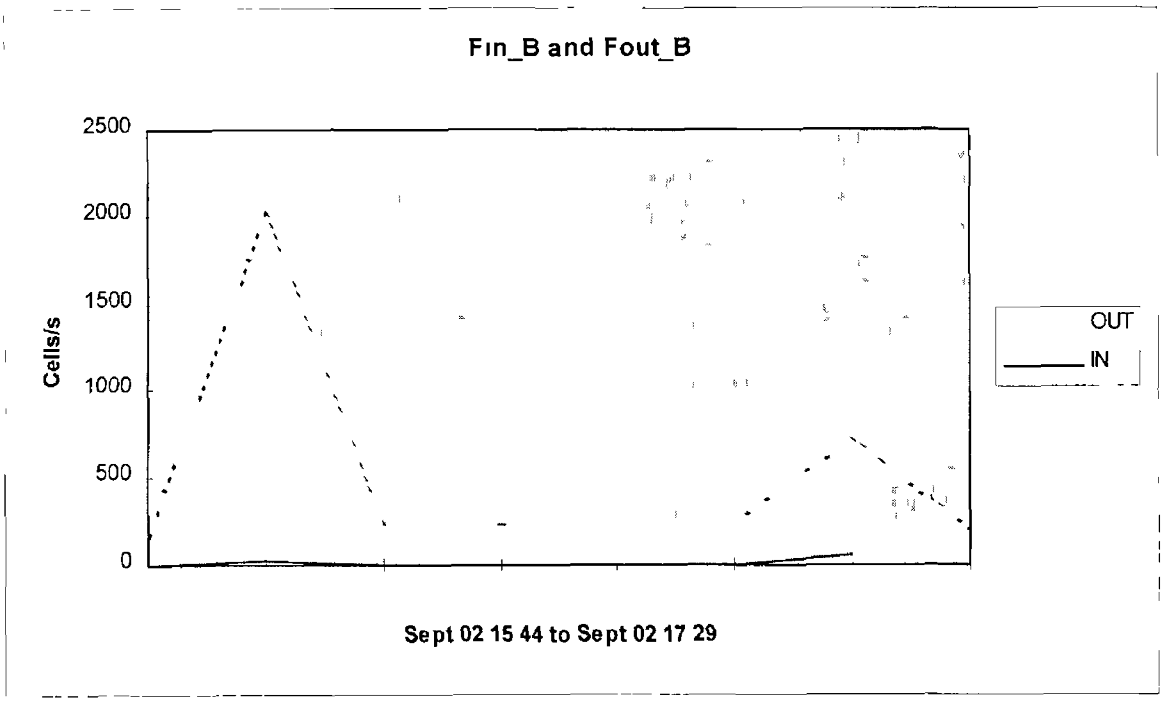


Fig A 3 2 FIONN recordings by NMS during Sept 02

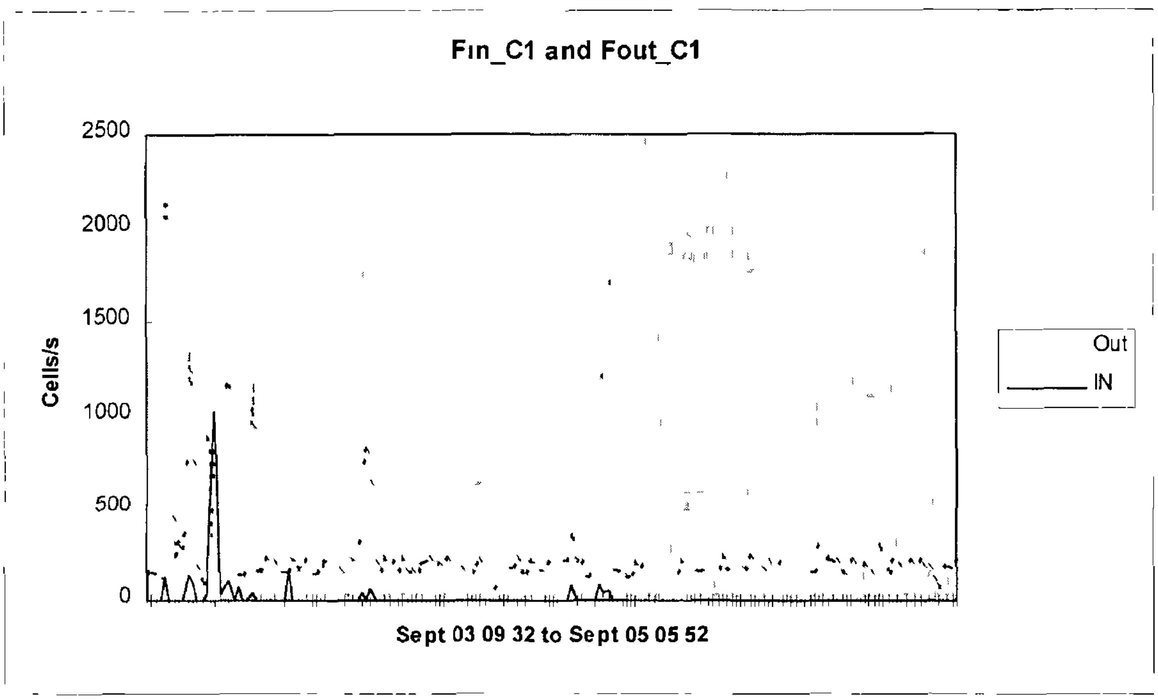


Fig A 3 3 FIONN recordings by NMS during Sept 03 and Sept 05

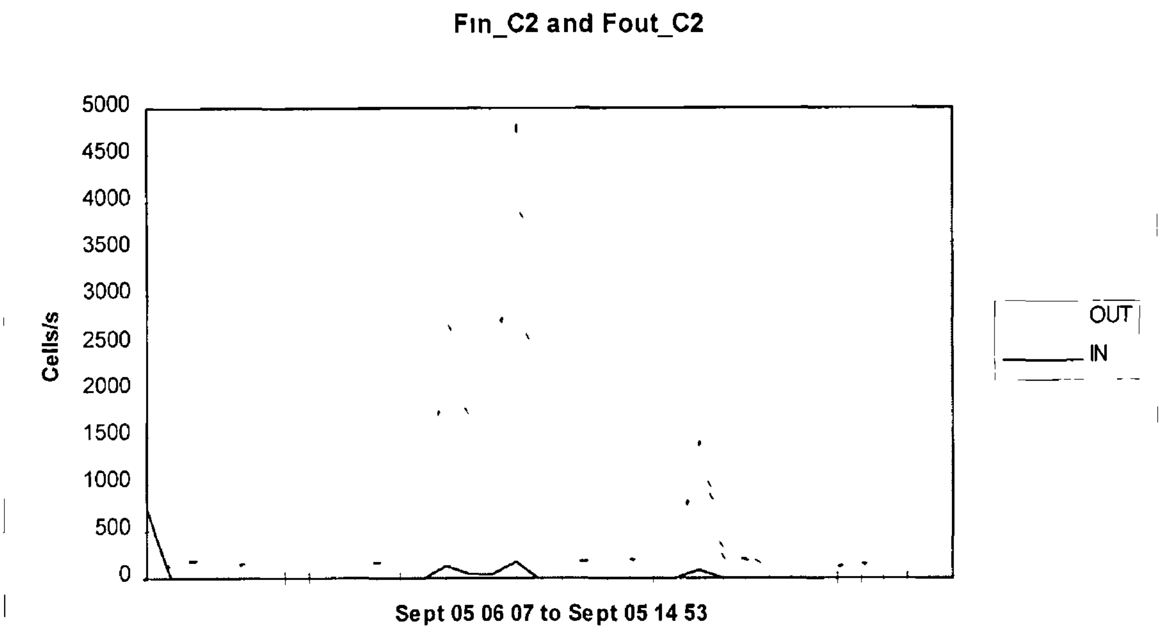


Fig A 3 4 FIONN recordings by NMS during Sept 05

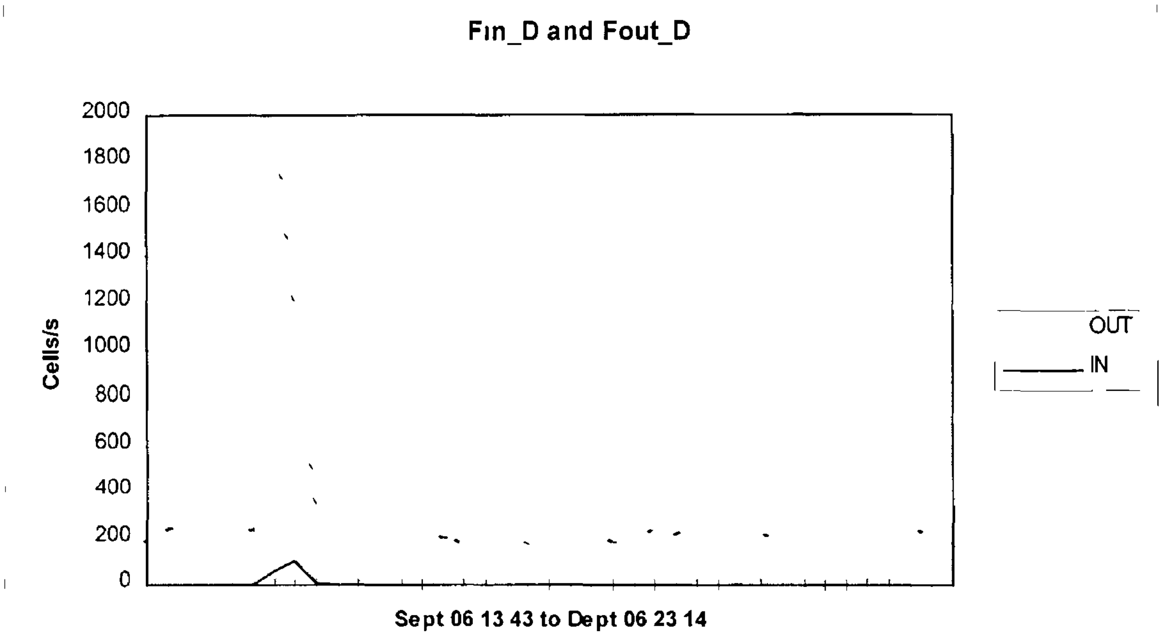


Fig A 3 5 FIONN recordings by NMS during Sept 06

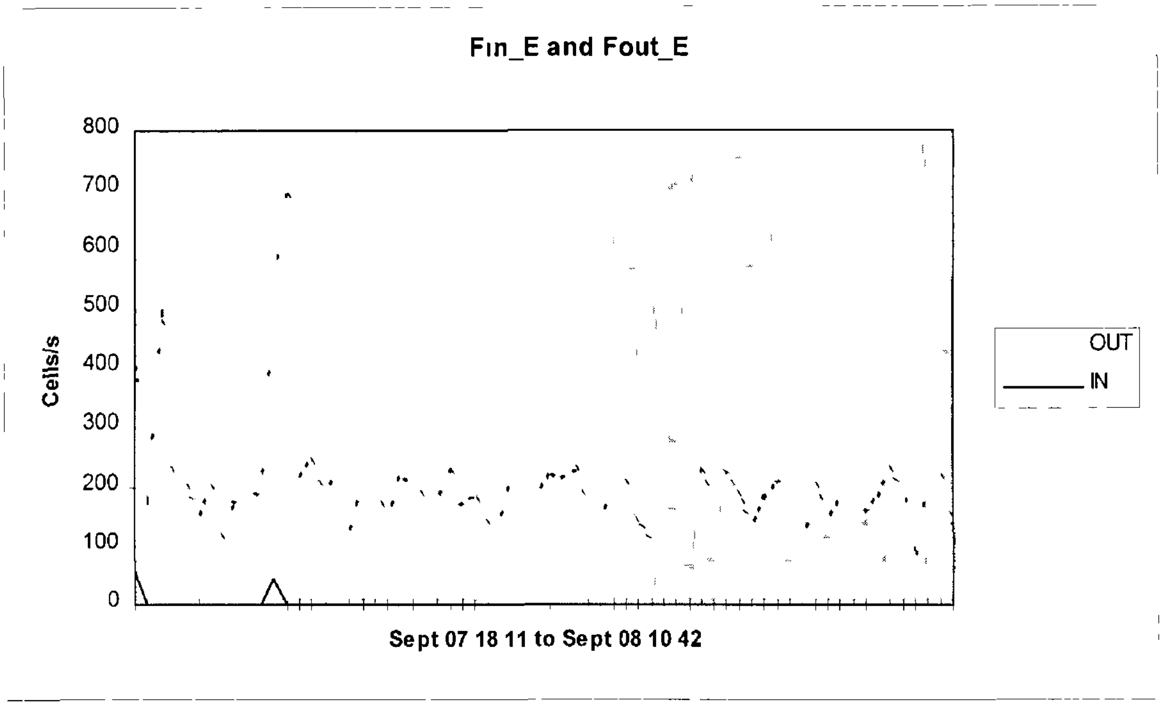


Fig A 3 6 FIONN recordings by NMS during Sept 07 and Sept 08

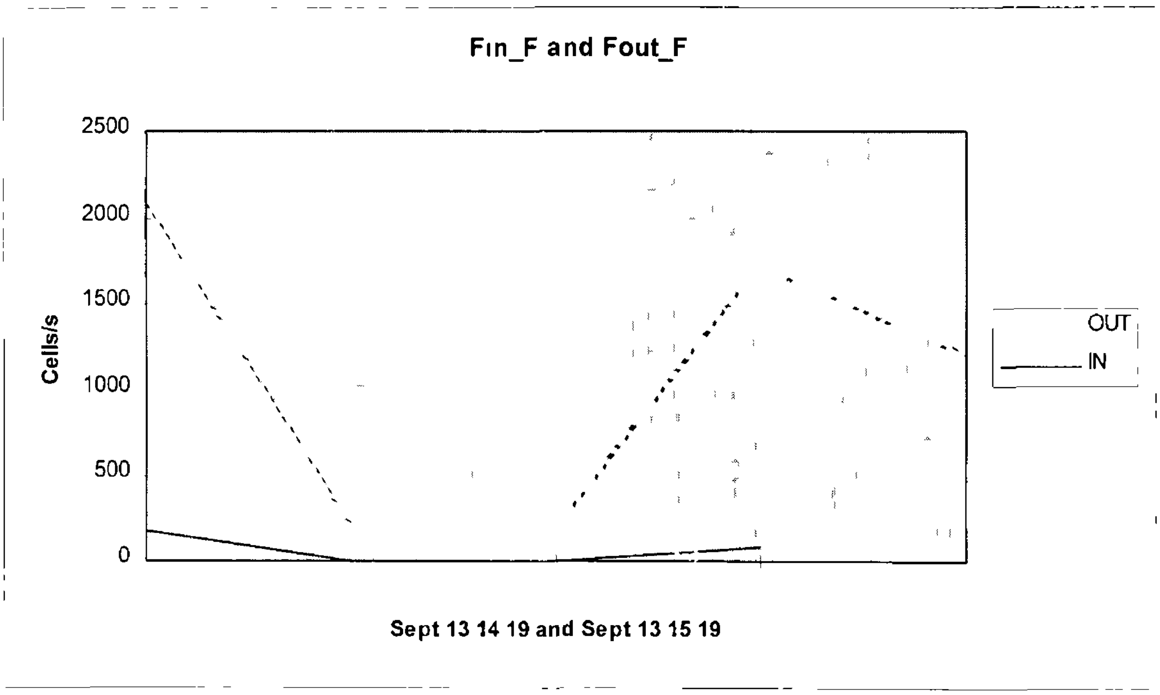


Fig A 3 7 FIONN recordings by NMS during Sept 13

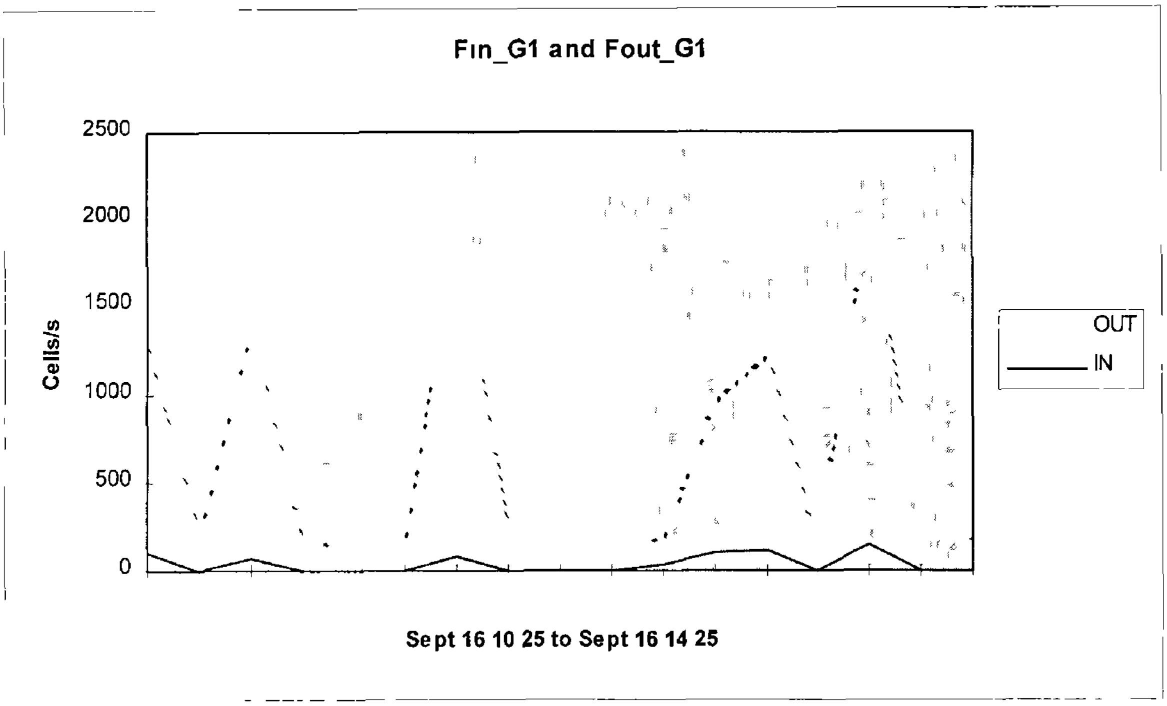


Fig A 3 8 FIONN recordings by NMS during Sept 16

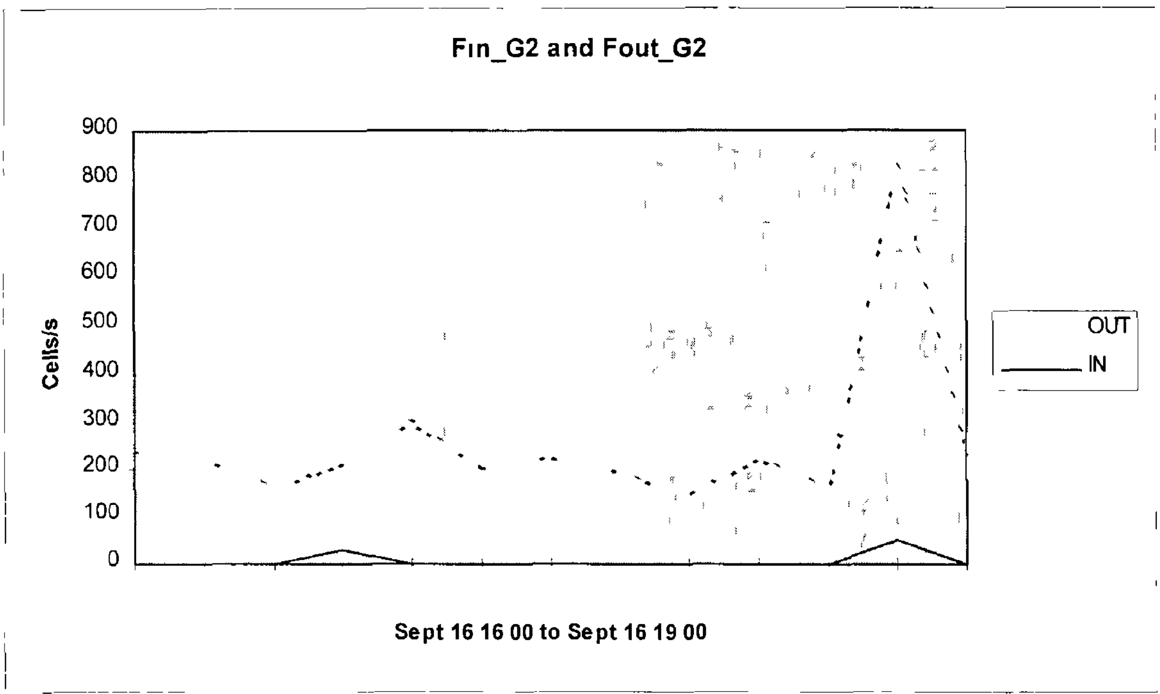


Fig A 3 9 FIONN recordings by NMS during Sept 16

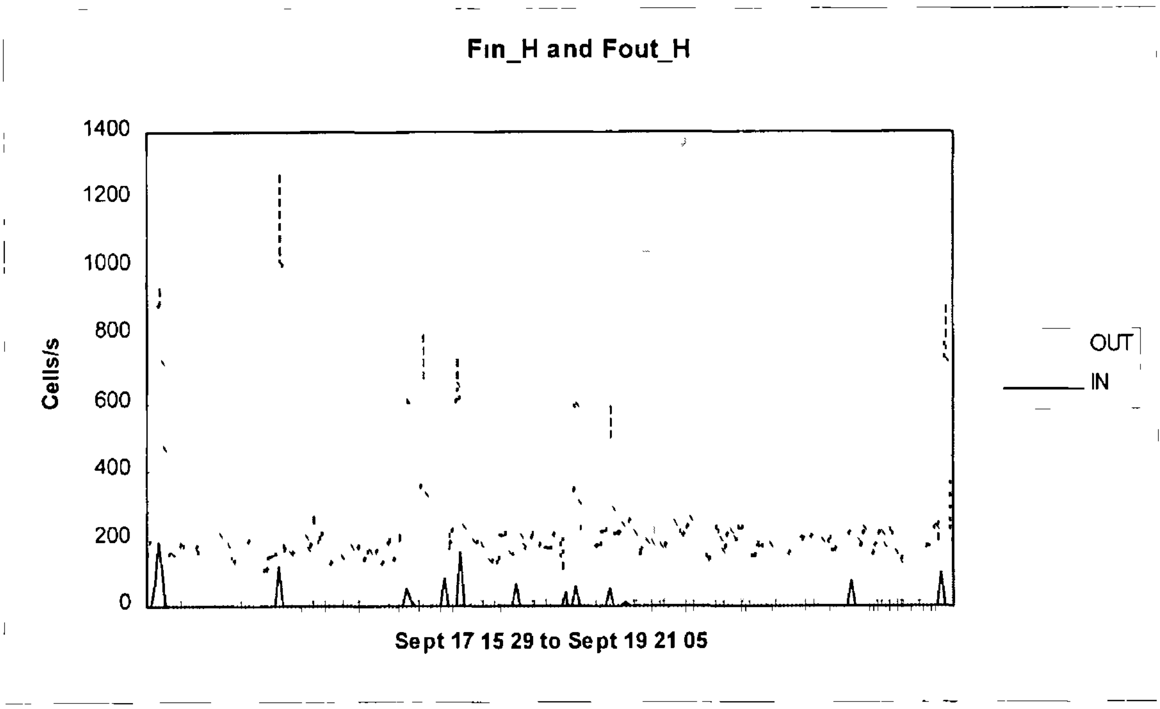


Fig A 3 10 FIONN recordings by NMS during Sept 17 and Sept 19

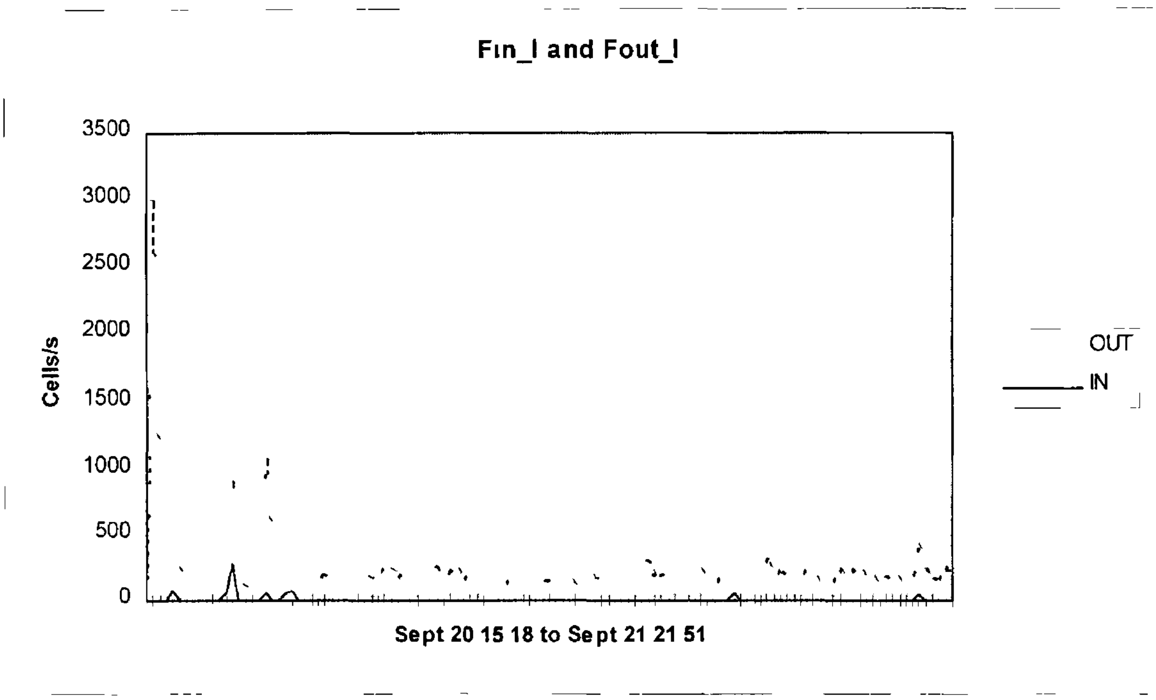


Fig A 3 11 FIONN recordings by NMS during Sept 20 and Sept 21

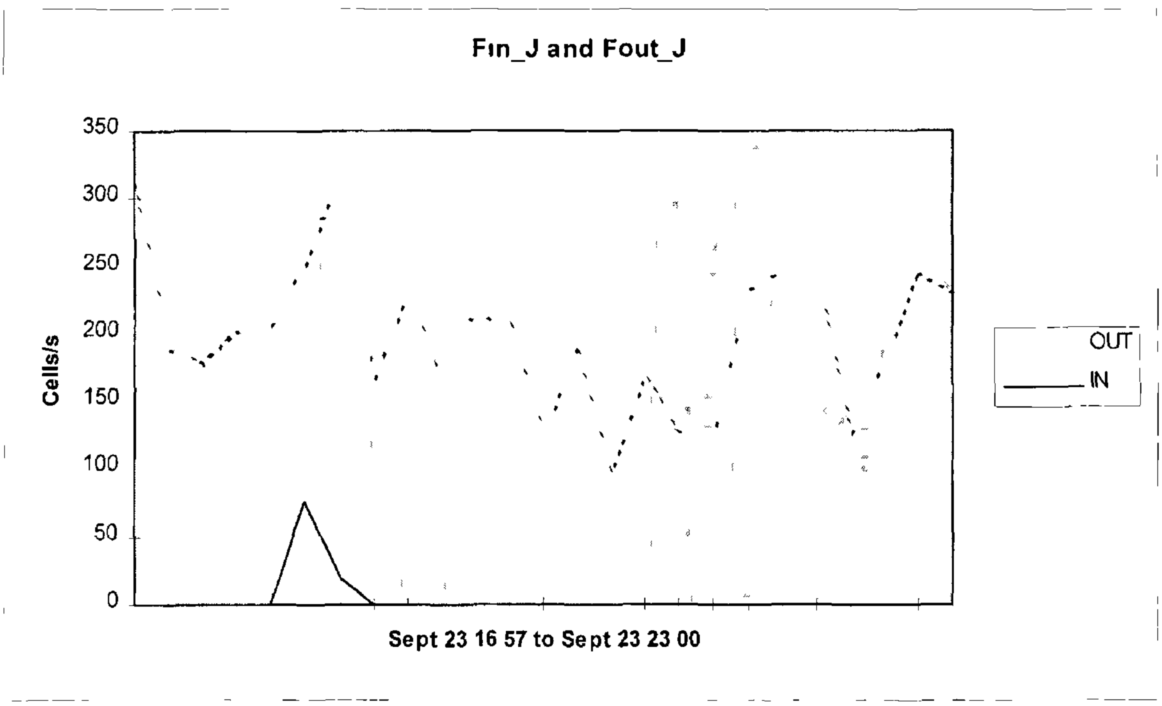


Fig A 3 12 FIONN recordings by NMS during Sept 23

Appendix D. List of Modified WWW Server Program

Http_log.c

```
void log_transaction()
{
        char    str[HUGE_STRING_LEN],
        long    timz,
        struct  tm * t,
        struct timeb mili,
        char    tstr[MAX_STRING_LEN],   sign,   buffer[MAX_STRING_LEN],
        filename[MAX_STRING_LEN],

        int     sockfd,
        struct sockaddr_un servaddr,

        ftime(&mili),
        t = localtime(&mili time),
        sprintf(buffer, " %03d", mili milhtm),

        if(timz < 0)
                timz = -timz,*/

        strftime(tstr, MAX_STRING_LEN, "%d/%b/%Y %H %M %S", t),
        strcat(tstr, buffer),
        sscanf(the_request, "%*s%s%*s", filename),
        sprintf(str, "%s [%s - %s] \"%s\" ", remote_name, start, tstr, filename),
        sprintf(str, "%s%d ", str, status),

        if(bytes_sent != -1)
                sprintf(str,"%s%d\n",str,bytes_sent),
```

Appendix D. List of Modified WWW Server Program

Http_log.c

```
void log_transaction()
{
	char	str[HUGE_STRING_LEN],
	long	timz,
	struct	tm * t,
	struct timeb mili,
	char	tstr[MAX_STRING_LEN],   sign,   buffer[MAX_STRING_LEN],
	filename[MAX_STRING_LEN],

	int	sockfd,
	struct sockaddr_un servaddr,

	ftime(&mili),
	t = localtime(&mili time),
	sprintf(buffer, " %03d", mili millitm),

	if(timz < 0)
		timz = -timz,*/

	strftime(tstr, MAX_STRING_LEN, "%d/%b/%Y %H %M %S", t),
	strcat(tstr, buffer),
	sscanf(the_request, "%*s%s%*s", filename),
	sprintf(str, "%s [%s - %s] \"%s\" ", remote_name, start, tstr, filename),
	sprintf(str, "%s%d ", str, status),

	if(bytes_sent != -1)
		sprintf(str,"%s%d\n",str,bytes_sent),
```

```
        else
                strcat(str,"-\n"),

        fprintf(xfer_log,"%s\n",str),
        fflush(xfer_log), */
        write(xfer_log,str,strlen(str)),

        sockfd = socket(AF_UNIX, SOCK_STREAM, 0),

        if (sockfd >=0)
        {
                strcpy(servaddr sun_path, "/tmp/anal"),
                servaddr sun_family = AF_UNIX,

                if (connect(sockfd, (struct sockaddr *) &servaddr, sizeof(servaddr)) >= 0
)
                        send(sockfd, str, strlen(str), 0),
        }
}
```

Http_request.c

```
void process_request(int in, FILE *out)
{
        struct timeb milli,
        struct tm * tmstruct,
        long timez,
        char milsec[MAX_STRING_LEN],

        get_remote_host(in),
        signal(SIGPIPE,send_fd_timed_out),
```

```
if(getline(as_requested,HUGE_STRING_LEN,in,timeout))
        return,
if(!as_requested[0])
        return,

strcpy(the_request, as_requested),

getword(method,as_requested,' '),
getword(args2,as_requested,' '),
getword(url2,args2,'?'),

unescape_url(url2),
getword(protocal,as_requested,'\0'),

if(protocal[0] != '\0')
{
        assbackwards = 0,
        get_mime_headers(in,out, url2),
}
else
        assbackwards = 1,

if (bytes_sent==-1)
{
        ftime(&milli),
        tmstruct = localtime(&milh time),
        strftime(start,     MAX_STRING_LEN,     "%d/%b/%Y %H %M %S",
tmstruct),
        sprintf(milsec, " %03d", milli millitm),
        strcat(start, milsec),
}

if(!strcmp(method,"HEAD"))
```

```
        {
                header_only=1,
                process_get(ın,out,method,url2,args2),
        }
        else ıf('strcmp(method,"GET"))
                process_get(ın,out,method,url2,args2),
        else ıf('strcmp(method,"POST"))
                post_node(url2,args2,ın,out),
        else ıf('strcmp(method,"PUT"))
                put_node(url2,args2,ın,out),
        else ıf('strcmp(method,"DELETE"))
                delete_node(url2,args2,ın,out),
        else
                dıe(BAD_REQUEST,"Invalıd or unsupported method ",out),
}
```

Appendix E. List of Charging Parameters

The following table summarises all the charging parameters so far identified It also includes (1) the entity responsible for the measurement of the corresponding charging parameters and (2) any special requirements identified with this measurement

Part	Parameter	Entity responsible	Special requirements
Access part	type of physical access	Operation System	
	length of access	Operation system	
	Maximum bandwidth allowed	Operation system User plane	UPC functions (user plane)
	Maximum number of simultaneous connections	Operation system Control plane	Limiting the number of on demand connections (control plane)
Core part (user plane)	Duration of the phase	Operation system Control plane User plane Management plane	Event notifications from the different planes to the operation system
	Distance (Location)	Operation system	Location of involved parties
	Traffic descriptor	Operation system User plane	UPC functions (user plane)
	Quality of service	Operation system Management Plane	F4/ F5 flows fully support
	Traffic volume used	Operation system User plane	Cell counting per connection at output links
	Time segment	Operation system	
	Number of tagged non conforming cells	Operation system User plane	Tagging and non conforming cell counting at UPC location
Core part (control plane)	Aspects related to the number of bandwidth modification requests	Operation system Control plane Management Layer	Bandwidth request notification from control to management plane
	Aspects related to the number of QoS class modification requests	Operation system Control plane Management plane	QoS request notification from control to management plane
	Aspects related to the number of add/ drop parties request	Operation system Control plane Management plane	Request notification from control to management plane
	Aspects related to the number of add/ drop connection request	Operation system Control plane Management plane	Request notification from control to management plane
Core part (management)	Establishment of connection	Operation system	
	Aspects related to the number of bandwidth modification requests	Operation system	
	Aspects related to the number of QoS class modification requests	Operation system	
	Aspects related to the number of add/ drop parties request	Operation system	
	Aspects related to the number of add/ drop connection request	Operation system	
	Rerouteing/ protection switching	Operation system Management plane	Ad hoc procedures for ATM protection switching