Billing Users and Pricing for TCP

Richard J. Edell, Nick McKeown and Pravin P. Varaiya

Abstract— **This paper presents a system for billing users for their TCP traffic. This is achieved by postponing the establishment of connections while the user is contacted, verifying in a secure way that they are prepared to pay. By presenting the user with cost and price information, the system can be used for cost recovery and to encourage efficient use of network resources. The system requires no changes to existing protocols or applications and can be used to recover costs between cooperating sites. Statistics collected from a four day trace of traffic between the University of California, Berkeley and the rest of the Internet demonstrate that such a billing system is practical and introduces acceptable latency. An implementation based on the BayBridge prototype router is described. Our study also indicates that pricing schemes may be used to control network congestion either by rescheduling time-insensitive traffic to a less expensive time of the day, or by smoothing packet transfers to reduce traffic peaks.**

Keywords— **Internet economics, network tracing, usage-accounting**

I. Introduction

SINCE its creation in 1986, the NSFNET has sustained an extremely rapid growth rate. With an estimated 20 million people reachable by electronic mail, the NSF funded backbone is serving a multitude of academic and commercial organizations. The NSF subsidy for NSFNET and various regional networks amounts to $20 million per year, of which about $12 million is for NSFNET. The total cost of the Internet is estimated to be about $200 million per year. It is likely that NSFNET will soon be gone, so that Internet will be run entirely by commercial and non-profit carriers, and all costs (and profits) will be recovered through charges and pricing schemes.

If all users within an organization generated similar amounts of traffic, cost recovery could be equitably and simply achieved by dividing the cost equally among users. However, as might be guessed, and as we shall demonstrate, different Internet users generate widely different amounts of traffic. Pricing schemes or charges that are designed to recover costs fairly from this diverse population of users and to allocate network resources in an efficient manner require the capability to meter individual user traffic and to present users with prices and charges in a way that encourages efficient network use [1]. In this paper, we present a billing system that can be used to meter users' wide area TCP traffic, involving the users in the decision to consume resources and making them accountable for the traffic that they generate.

A number of network billing systems have been previously proposed [2],[3]; in addition, one billing system has been implemented [4]. An excellent discussion of methods, costs and benefits of usage metering as well as a proposed billing infrastructure is given in [2]. In [3], a flow-based accounting mechanism is defined based on IP data that flows between end hosts. Brownlee reports experiences with implementing a network billing system [4]. But none of these systems provide a mechanism for identifying and involving individual users; i.e. these systems identify hosts machines but not the users of those machines. In 1990, Estrin et al. suggest some research topics towards usage billing and feedback [5]. In particular, they propose the instrumentation of current networks to determine the performance implications of usage metering, and highlight the problem of identifying and authenticating the end user.

Our billing system is designed to meter users (i.e. the persons using hosts) of the TCP connection-oriented protocol [6],[7], which represents almost all of the traffic on the Internet. It is because a connection is established prior to the transfer of data that we are able to identify reliably the originating user and verify that they are prepared to pay for their traffic. An important feature of this billing system is that this can take place without modification to the TCP/IP protocols, or applications.

A usage-based billing system in which the user is directly involved will affect the way in which the Internet is used. Several schemes have been proposed for controlling congestion via pricing which require that traffic is metered and involve the user in the feedback mechanism [8]. However, it has not been previously shown that real-time metering of traffic for high-speed networks is feasible, or that it is practical to involve the user in the feedback.

A billing system that controls the access of individual users will also affect the way in which the Internet grows. Users within an organization who need a higher bandwidth connection to the commercial data carrier can pay for and get preferential use of the improved service.

Usage billing for network traffic is a controversial subject [6],[9]. It is not the purpose of this paper to argue for the widespread adoption of usage billing. Our aim is to demonstrate that it is technically feasible to implement a billing system that involves users and makes them accountable for their traffic.

In Section II, we provide a detailed description of the proposed billing system. In Section III, we describe an extensive feasibility study of the system, based on a trace of the traffic that leaves and enters the Berkeley campus. Section IV shows how the BayBridge, a prototype router,

Manuscript received November 1, 1994; revised April 15, 1995. This research was funded by grants from Pacific Bell, the California State MICRO Program 93-152 and NSF grant IRI-9120074.

R. Edell and P. P. Varaiya are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, California 94720 USA. Their email addresses are {edell,varaiya}@eecs.berkeley.edu.

N. McKeown is with the Departments of Electrical Engineering and Computer Science, Stanford University, Stanford, CA 94305 USA. His email address is nickm@ee.stanford.edu.

could be to used to meter traffic. Section V discusses some other relevant measurements. We draw some conclusions from our study in Section VI.

II. Generic System Description

A number of features are essential to any billing system:

- *No changes to existing Internet protocols* — Because of the huge number of installed end systems, bridges and routers, it is important that an Internet billing system work with the existing Internet protocols. It means that the billing system should not require the use of any special option fields by end systems (for example, IP options or TCP options).
- *No changes to existing applications* — Many applications such as ftp, email, gopher and mosaic are in widespread use today and collectively contribute to most of the traffic on the Internet [10]. A billing system should be able to meter traffic for these and other existing applications without change.
- *User Involvement* — If a billing system charges individual users for their traffic, it must first determine the identity of the user. It should also obtain the explicit approval from the user or an authorized agent that they will pay for the resources consumed. For security and credibility, this approval should be authenticated, for example with Kerberos [11]. And finally, the billing system should provide accurate and credible on-line feedback to the user as they consume resources. This implies that the metering of traffic should be exact and not based on traffic sampling.

It would be desirable for a billing system to have the following additional features:

- *Provide on-line reporting of aggregate network usage* — This enables schemes that control network congestion based on global traffic measures. The control may be implemented using priorities or by a time varying pricing policy [8].
- *Allow continued sharing of information and resources* — The growth of the Internet can be attributed to applications that encourage the sharing of information and resources between remote sites. It would be advantageous if billing systems could cooperate to identify the user and bill them for their traffic.

A. Detailed Description of Billing System

A block diagram of the billing system is shown in Figure 1. This figure shows a single administrative domain that is connected to the outside world via a Billing Gateway (BGW) and a billed link. The administrative domain contains a collection of networks, users and hosts. The billing system controls users' access to the billed link by allowing or disallowing TCP connections and by metering users' TCP traffic once a connection has been established.

The basic operation is as follows: when a user attempts to establish a TCP connection with the outside world, the BGW postpones the establishment of the connection while it tries to identify the originating user. The system contacts the user, verifying that they want to establish the connection and that they are prepared to pay for it. If the user accepts the connection, the normal connection establishment is allowed to continue and the BGW will begin metering this connection's traffic.

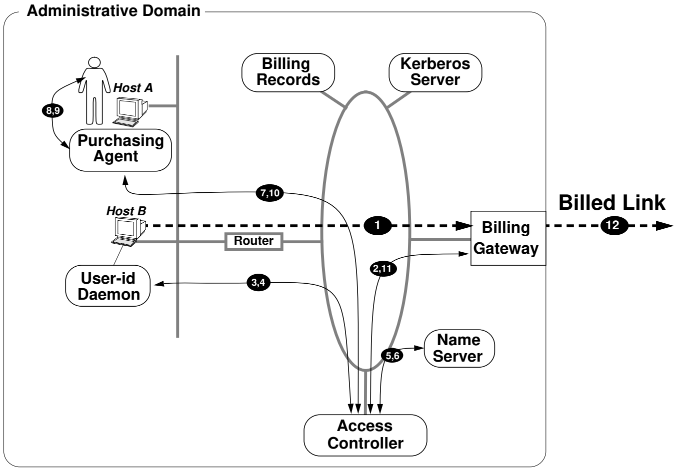


Fig. 1. Components of the billing system showing the major communications required to establish a connection.

We will describe the operation in detail by way of an example and by referring to the communications marked on Figure 1 and the corresponding time diagram in Figure 2b. The timeline in Figure 2b is an extension of Figure 2a showing all the communications involved in setting up a connection between the originating local host and the remote host.

In the example, the user sits at Host A but is logged in to Host B. The user's application on Host B attempts to establish a TCP connection to a remote host. Considering each communication in turn:

1. Host B initiates the connection by sending a TCP SYN message. The BGW recognizes the TCP SYN message as an attempt to establish a new connection to the outside world. The BGW holds onto the TCP SYN message while it determines whether or not the connection should be allowed and access granted to the billed link. This is achieved by communications (2-11) to identify and contact the user, determine whether the connection will be allowed and to set up the necessary state for metering and billing records. The connection is always referred to by its unique identifier: [(source (address,port), destination (address,port)].
2. The BGW contacts the Access Controller which is responsible for determining whether the connection should be allowed. It achieves this by communicating with each of the components in turn.

3,4. The Access Controller asks the User Identification Daemon (Useridd) to identify the user.

5,6. The Access Controller asks the User Name Server to locate the user's Purchasing Agent.

7. The Access Controller asks the user's Purchasing Agent to verify that the user wishes to establish and pay for this connection.

8,9. The Purchasing Agent may be configured to respond

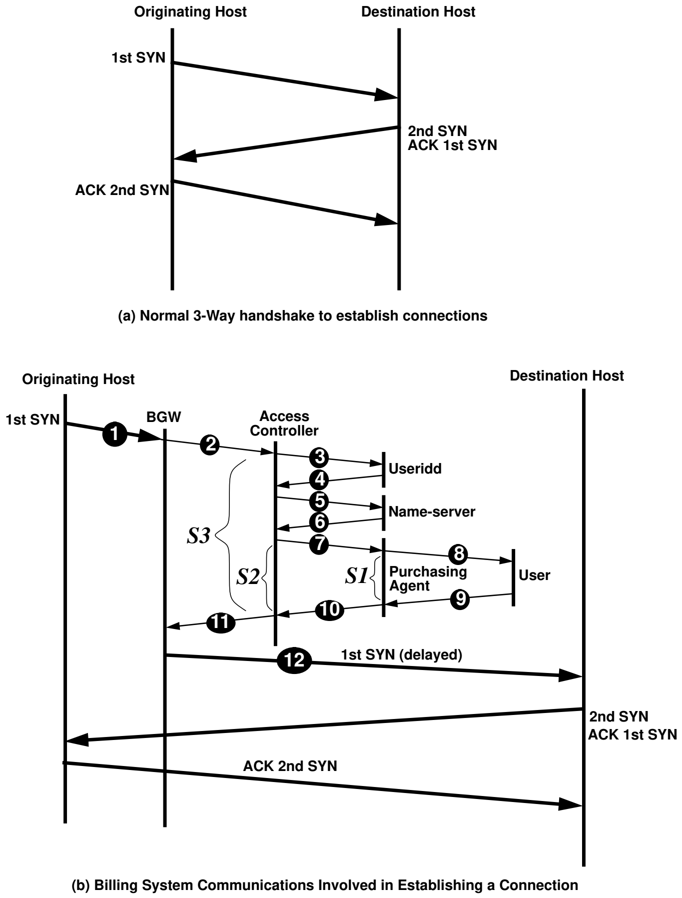


Fig. 2. The communications required to establish a connection without and with the billing system.

automatically on behalf of the user, or may request the explicit authorization of the user by means of a dialog box on the user's screen.

10. The Purchasing Agent responds to the Access Controller authenticating its reply using Kerberos.
11. If the user confirms that they wish to establish and pay for the connection, the Access Controller tells the BGW to allow access to the billed link and meter the connection.
12. The BGW creates an entry for this connection in its connection tables and forwards the TCP SYN message towards the remote host.

Now that the BGW knows to allow this connection, all future messages for this connection in either direction will be forwarded without further delay. This means that when remote host responds with a TCP SYN+ACK, the BGW will forward the message, updating its connection table entry.

B. Components

The components in Figure 1 fall into three categories. The first category is the Access Controller which was described in detail above.

The second category is user involvement. This contains three components: User Identification Daemon, User Name Server and Purchasing Agent. Together these components identify the originating user, obtain verification from the user and provide feedback to the user as resources are used.

- The User Identification Daemon (Useridd) is a system daemon process that runs on all multi-user machines within the administrative domain. Request messages are sent to the Useridd containing the unique connection identifier. The Useridd examines kernel data structures to determine and return the name of the user who originated the connection.
- The User Name Server is a standard network naming service which when supplied with a user name will return the location of the user's Purchasing Agent.
- The Purchasing Agent[1] is a user-level process responsible for verifying that the user is prepared to pay for the TCP connection. In its simplest form, the Purchasing Agent could just pop up a dialog box on the user's screen asking for a simple accept/reject response. Alternatively it may be configured to automatically respond to some or all of the requests that it receives from the Access Controller. For example, it could be configured by the user to: (i) automatically accept all connections below a certain price, but verify with the user before accepting more expensive connections; (ii) automatically accept all connections to specific well-known ports, but verify with the user before accepting others; or (iii) automatically accept all connections to certain destinations. The Purchasing Agent proves its authority to represent the user by including a Kerberos authenticator in its response.

The third category is metering. The two components in this category are the BGW and Billing Records.

- The BGW is a specialized router that maintains a table of established TCP connections for metering traffic, in addition to performing the usual IP routing functions. The BGW must understand TCP connections so that it can determine which connection record to update. New connections are TCP messages for any connection that the BGW has not seen before. This includes connections that establish via other BGWs.
  The BGW must determine when connections have closed so that the entry in the connection table can be freed. Connections may close in a number of different ways and the TCP FIN messages that close the connection may travel via a different BGW. So that connections can be removed from the tables in a timely manner, the BGW times out connections that are inactive for a long period. If the connections become active again, they are added back into the table.The BGW maintains a table of all connections with a separate entry for each established connection keyed by its unique connection identifier. This means that the connection tables maintained by the BGW may be much larger than IP routing tables. It also means that the key into the connection table is much larger: for an IP router the key would normally just be the 32-bit destination network address. For a BGW with TCP/IP the key is 96-bits. In Section III.D we will consider how

[1] For security purposes, this process is usually run on the host at which the user sits.

large the connection tables need to be and compare several schemes for looking up entries in the connection tables.

- The Billing Records are responsible for maintaining records of metered connections and for providing on-line feedback to the user as the connection progresses.

C. Simpler Cases

A number of possible simplifications could be used to reduce the number of communications described above. Each simplification corresponds to bypassing communications shown in Figure 2b, "S1" - "S3". The simplifications reduce the number of communications involved in establishing a connection, thus reducing the latency seen by the user.

1. *No user interaction* — Do not explicitly involve the user in the connection acceptance. This was discussed above and would be achieved by configuring the Purchasing Agent to accept connections automatically. The advantages are that it would make the connection setup faster and would not burden the user with involvement in every connection.
2. *No Purchasing Agent interaction* — Configure the Access Controller so that connections from some users or for some applications are accepted automatically. This eliminates communications (5-10).
3. *By machine billing* — Configure the Access Controller so that connections from some machines are accepted automatically. This would be used when a machine is used by a single user, or if it has been agreed that the traffic will always be paid for regardless of the user. In fact, this would be necessary for machines that cannot support the Useridd process (this would include single-process personal computers). In this case, we eliminate communications (3-10).

It is important to note that both simplifications 2 and 3 compromise the security of the billing system as neither obtains authenticated feedback from the user.

D. Complications & Gotchas

We believe that the billing system described above is feasible and implementable. However, there are several things that can make the implementation more difficult, or make the billing model less appropriate. We consider some of these below.

- *IP Fragmentation* — IP datagrams may be fragmented by routers as they progress through the network. The BGW must interpret TCP messages and so it must be able to reassemble fragmented IP datagrams. In our feasibility study, we found that less than 0.01% of our wide-area datagrams are fragmented[2].
- *Malicious IP Fragmentation* — IP datagrams may also be severely fragmented by the end station in an attempt to circumvent the billing system. This may be overcome by refusing to forward fragmented datagrams with too few data bytes.

[2] All of the machines on the Berkeley campus that fragment their outgoing traffic are Macintoshes.

- *Hiding data in option fields* — To avoid being billed, the user may attempt to hide data in IP or TCP options fields. This may be overcome by metering protocol header bytes as well as data bytes. This way, anomalies can be detected and billed for accordingly.
- *Time varying route* — The packet-switched Internet allows datagrams within a connection to be delivered via different routes. If the change is due to load balancing, we can expect the price for delivery to be unchanged. But if there is a change in topology, the price may change and should be communicated back to the user.
- *Connections on behalf of, but not originated by a user* — The main reason for involving the user is to make them accountable for their traffic. The basic system achieves this by billing the originator of a TCP connection. However, it would be more appropriate to bill the beneficiary of the connection. In practice, the *originator* of a connection is not necessarily the *beneficiary*. For example, if a process on one host originates a connection so that it can transfer data to or from a remote host, who is the beneficiary of the transfer?
  We discuss a solution to this problem below when we describe how cooperative billing can be achieved.

E. Enhancements to the Basic Billing System

- *Cooperative Billing Between Administrative Domains* — If two Administrative Domains wish to bill for a connection that flows between them, they must first agree which domain will pay for the connection. That domain then needs to determine which of its users to bill. The user that is billed should be the beneficiary of the connection.
  The following method could be used to determine the beneficiary. When the connection is established, the BGW asks the originating user if they will pay for the connection. If the user is the beneficiary, they should agree to pay. Alternatively, the user may request that the remote end pay. In this case the local BGW contacts the remote BGW which in turn contacts the remote user asking if they are prepared to pay. If neither user agrees to pay, the connection is refused. If either of the end points of the connection is a service, rather than a user, its local Access Controller could be configured to accept connections on its behalf.
- *User Specified Limits* — The Purchasing Agent could respond to the Access Controller with conditions and limits to its acceptance. For example, the user may be willing to pay up to a specified amount for a connection, or for a specified number of packets or bytes. The user may only want to pay for traffic as long as the price for the connection stays below a specified threshold. This would allow the implementation of a time-varying threshold pricing policy agreed upon by the user and policed by the BGW [8]. If during the life of the connection any of the limits or thresholds are exceeded, the BGW contacts the user before forwarding any more messages.

- *Metering UDP Traffic* — Our billing system is designed to meter TCP traffic, which currently comprises almost all of the traffic on the Internet. However, a growing proportion of traffic is using the UDP protocol, e.g. MBONE. In principle, our billing system could also meter traffic from these applications. There are two principal requirements. First, that prior to the BGW forwarding datagrams it must be possible to identify the user in an equivalent way to the User Identification Daemon. Second, that it be possible for the BGW to relate each datagram to a metered flow. How practical this is will depend on the application.

III. Feasibility Study

In this section, we will use the results from a detailed study to demonstrate that our billing system is feasible even for large campus networks. Our results are derived from a trace of network traffic on the Berkeley campus FDDI backbone network.[3]

Most of our billing system's complexity is connection related. Therefore, our feasibility study measures several key statistics about TCP connections. These statistics are: connection setup time, connection setup intensity, active connection count, and complexity of searching the connection table. Under our billing system, connection establishment is delayed while locating the user and then verifying that they will pay; therefore, we measured connection setup *latency* without the billing system and estimate the additional connection setup latency introduced by the billing system. Several components of our billing system are activated once for every connection; therefore, we measured connection setup intensity to determine the required *throughput* for these components. The BGW maintains a table of active connections; therefore, we measure the evolution of connection count to determine the required *size* of the connection table. The connection table is frequently searched; therefore, we measure the performance of different search methods to determine the *complexity* of connection table lookup.

The BGW processes every datagram for metering; therefore, our feasibility study measures datagram intensity to determine the required datagram *throughput* for the BGW.

A. About the Trace

The Berkeley campus community is composed of 40,000 students, faculty and staff members. The campus network interconnects 22,000 hosts of which 6,312 hosts participated in WAN TCP connections during the study period. The campus network is connected to the Internet by two 1.5Mbps "T-1" links and a single boundary router. We consider everything on the other side of this router to be the "outside world."[4]

[3] During the month of September 1994, Berkeley contributed the 13th largest amount of WAN traffic to the NSFNET out of over 22,000 registered networks [12].

[4] This definition of "outside world" includes some geographically and topologically nearby sites such as Lawrence Berkeley Laboratories.

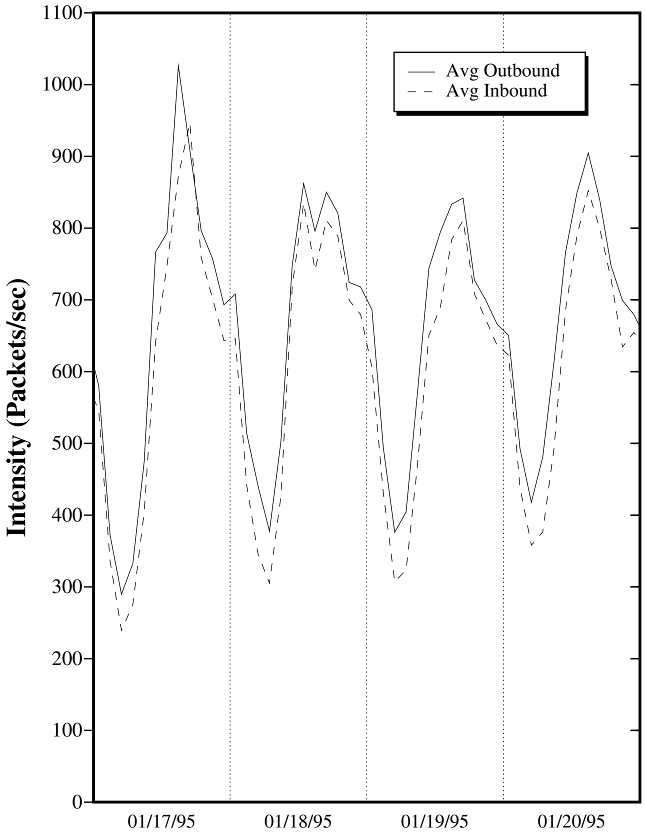


Fig. 3. Average intensity of datagrams entering and leaving the Berkeley campus.

Our trace was obtained by logging all TCP/IP headers that appeared on the Berkeley backbone network. A 50ns timestamp is attached to each log entry. The analysis presented in this paper considers WAN traffic over the four days from midnight Monday January 16 through midnight Friday January 20, 1995. During our study period, we measured 89 gigabytes of WAN traffic in 439 million datagrams. Of this WAN traffic, 92% of the bytes and 92% of the datagrams were for TCP traffic. Of this WAN TCP traffic, 86% of the bytes and 84% of the datagrams were from multi-user hosts (see Table I). Figure 3 shows how the average rate of inbound and outbound datagrams changed over our study period.

B. Additional Connection Setup Latency

Figure 2a illustrated the communication involved in establishing a TCP connection without billing and Figure 2b illustrated the additional communication required by our billing system. Obviously this additional communication increases connection setup time. The relative significance of this additional delay depends on the distribution of connection setup times. Figure 4 shows the cumulative distribution of measured connection setup times for outbound connections without the billing system. Our analysis showed that 80% of the connections took between 24ms and 734ms to establish, with a median time of 116ms, and an average time of 519ms.

TABLE I
AMOUNT OF TCP WAN TRAFFIC BY HOST TYPE.

Host Type	Number of Hosts	Per Host Averages[a] Connections	Datagrams	Bytes	Percentage in Category Datagrams	Bytes
Multi-user Computers	2,119	1,306	153,226	32,428,047	83.8%	85.7%
Personal Computers	1,527	64	5,447	1,167,732	2.1%	2.2%
Unknown	2,666	237	20,413	3,617,429	14.0%	12.0%

[a]These are four day averages.

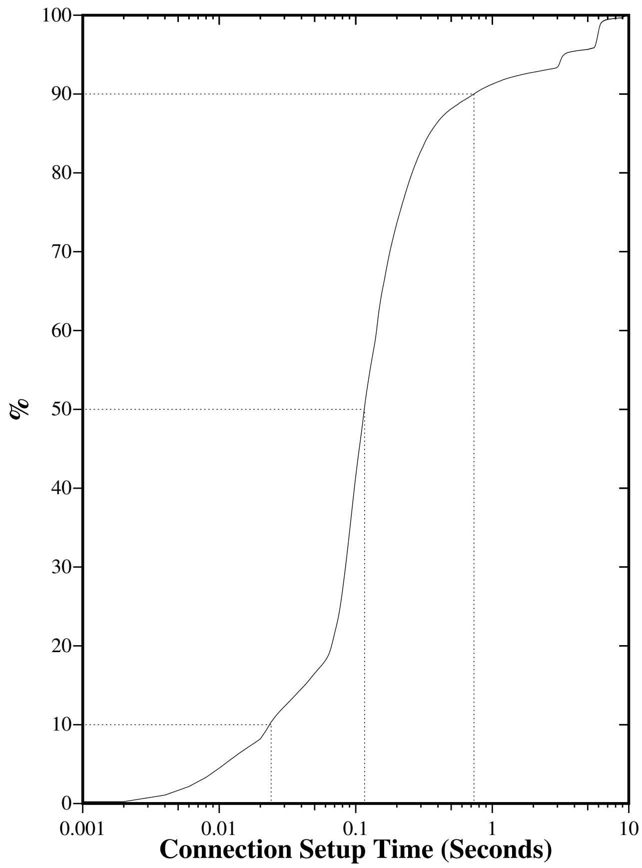


Fig. 4. Cumulative distribution of setup latencies for connections that originate on the Berkeley campus and connect to a host elsewhere on the Internet.

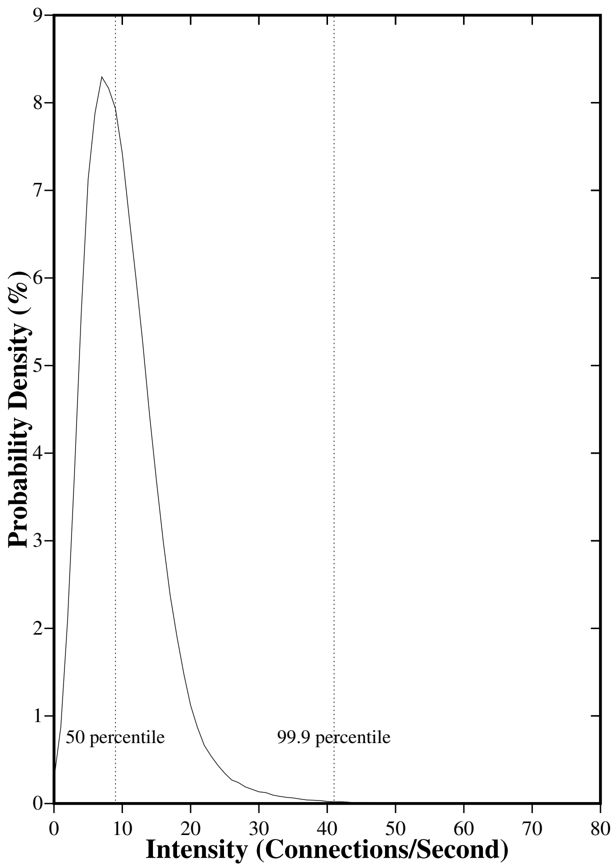


Fig. 5. Rate at which connection are established to and from the Berkeley campus.

Our trace experiment included the identification of users on selected machines using the "User Identification Daemon" (Useridd) implemented as a user-level process. Our analysis[5] showed that the average response time was 84ms.

We can use these results to estimate how long it would take for the rest of the connection setup overhead. Let's assume that each of the other three request/reply pairs will take approximately the same amount of time. Then the average additional delay is 336ms. Therefore, the average connection setup time with the billing system would be 855ms, an increase of approximately 65%.

[5]This data is from an earlier trace recorded from midnight November 10 through midnight November 14, 1994.

C. Connection Throughput Requirements

This section discusses the throughput requirements for the functional blocks. Most of the billing system functional blocks are activated once per connection setup. Therefore, the peak intensity of connection setup is an important parameter. Figure 5 shows the distribution of connection setup intensities over one second intervals. The peak rate was 79 connection setups per second, and 99.9% of seconds had 41 or fewer connection setups. Therefore, the throughput requirements for the functional blocks can be easily achieved.

D. Size of Connection Table

While a connection is active, an entry must be maintained in the BGW connection table. For the billing system

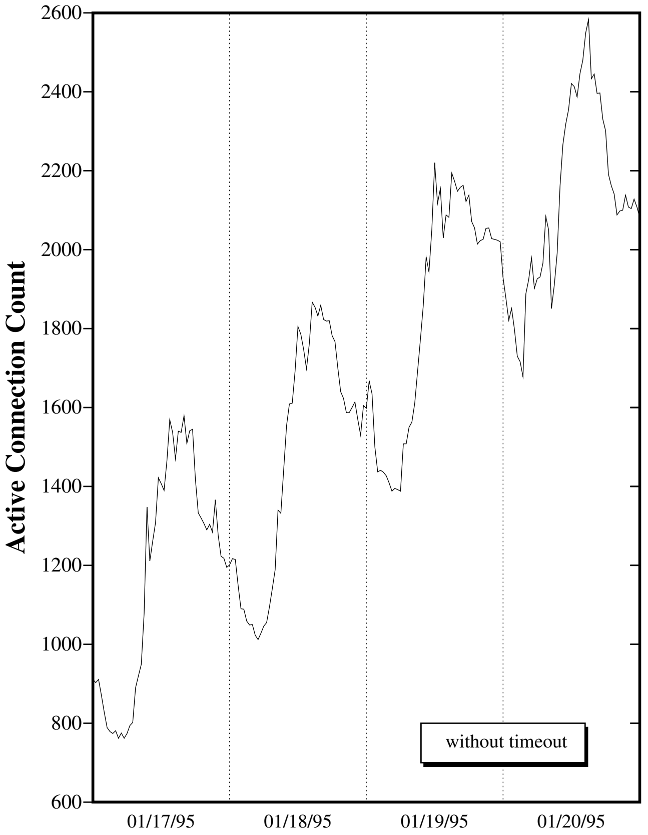


Fig. 6. A count of the number of currently established connections to and from the Berkeley campus.

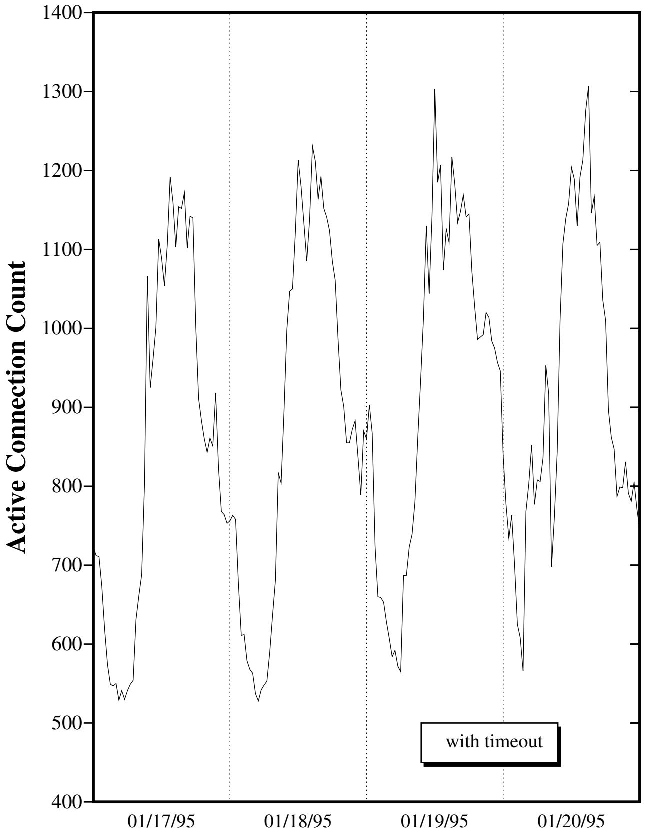


Fig. 7. A count of the number of currently established connections. Connections are timed-out after an inactivity period of 60 minutes.

to be feasible, these tables must not be too large. We have used our trace data to reconstruct the connection count evolution in two different ways.

The first reconstruction method removes connections from the table only when the trace data indicates that a connection has closed. Figure 6 shows this reconstruction. This figure clearly exhibits an upward drift in the connection count. Some of this drift can be attributed to long-duration connections that start but do not end during the trace period. Most of the drift is from hosts that fail to properly close-down connections. From this, we conclude that a BGW can not reliably detect the closing of all connections.

The second reconstruction method incorporates an inactivity time-out. Figure 7 shows the evolution of connection count if connections are timed-out after 60 minutes. With the time-out policy the maximum connection count is below 1,350 entries and there is no longer any significant upward drift.

E. Complexity of Connection Lookup

The BGW meters every TCP message that it forwards. It is therefore important that connection table entries are located quickly. These entries are keyed by a 96-bit connection identifier. Using this key to directly index a 2^{96} entry table in RAM or using a 96-bit wide CAM (contents addressable memory, sometimes called associative memory) is impractical.

We have found effective hashing functions for hardware and software implementations using either CAM or RAM based connection tables. We emulate the behavior of each hashing function with our trace data, measuring the average number of lookups that are required to find the correct table entry.

For CAM based tables, we considered three widths that are commonly available and used in commercial IP routers: 48, 32 and 16 bits wide. A hashing function is applied to the 96-bit key to reduce it to a 48, 32 or 16-bit index which is used to find the first match in the CAM. Associated with each CAM entry is a shadow entry at the same offset in RAM. The shadow entry contains an orthogonal portion of the key that when combined with the index can be used to recover the full 96-bit key. The orthogonal portions of the key are compared to verify that the entries match. If they do not, then the search is repeated looking for the next match in the CAM. Figure 8 illustrates this structure.

Figure 9 shows how CAM hashing functions $f_{CAM,1}$, $f_{CAM,2}$, $f_{CAM,3}$ map the 96-bit connection identifier to 48, 32 or 16 bits. Figures 10a-c plot the average number of comparisons per table lookup over the trace period. We see that function $f_{CAM,1}$ is much less effective than the others. This is because the port numbers for ftp and ftp-data differ by exactly one. Functions $f_{CAM,2}$ and $f_{CAM,3}$ both overcome this problem. Figure 10c shows that the perfor-

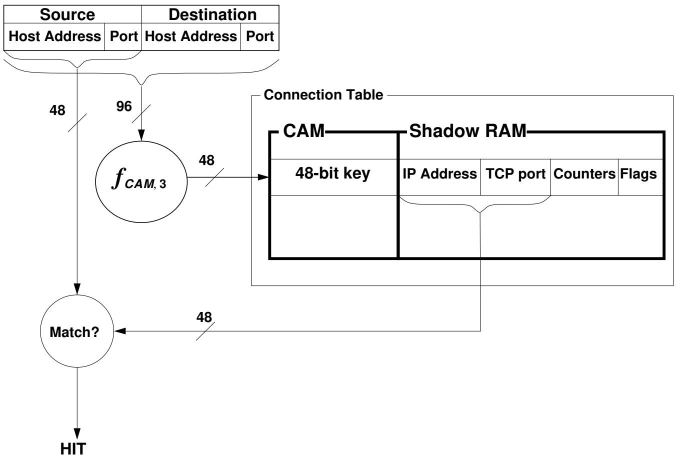


Fig. 8. The structure of the CAM-based connection tables in the BayBridge prototype.

CAM Width (bits)	CAM Hash Function $f_{CAM,1}$	CAM Hash Function $f_{CAM,2}$	CAM Hash Function $f_{CAM,3}$
48	$A_0 \oplus A_1$ (32), $P_0 \oplus P_1$ (16)	$A_0 \oplus A_1$ (32), $\overline{P_0} \oplus P_1$ (16)	$\overline{A_0} \oplus A_1$ (32), $\overline{P_0} \oplus P_1$ (16)
32	$A_0 \oplus A_1 \oplus P_0 \oplus P_1$	$A_0 \oplus A_1 \oplus \overline{P_0} \oplus P_1$	$\overline{A_0} \oplus A_1 \oplus \overline{P_0} \oplus P_1$
16	$\lfloor A_0 \oplus A_1 \rfloor_{16} \oplus P_0 \oplus P_1$	$\lfloor A_0 \oplus A_1 \rfloor_{16} \oplus \overline{P_0} \oplus P_1$	$\lfloor \overline{A_0} \oplus A_1 \rfloor_{16} \oplus \overline{P_0} \oplus P_1$

	RAM Hash Function $f_{RAM,1}$	RAM Hash Function $f_{RAM,2}$
RAM	$\lfloor A_0 \oplus A_1 \oplus \overline{P_0} \oplus P_1 \rfloor_{indexSize}$	CRC-32

Key to symbols: $A_0 \equiv$ IP address; $\overline{A_0} \equiv$ 16-bit swapped; $A \oplus B \equiv$ A *EXOR* B; $P_0 \equiv$ TCP port; $\overline{P_0} \equiv$ byte swapped; $\lfloor A \rfloor_B \equiv$ B least significant bits of A

Fig. 9. Hashing functions for CAM-based and RAM-based connection tables.

mance of these hashing functions is significantly degraded for a CAM width of just 16-bits.

We now turn our attention to the RAM based implementation. A connection table is implemented in RAM by applying a hashing function to the 96-bit key to generate a much smaller index. This index is used as the direct address of a hash-bucket in RAM. If the bucket contains more than one entry, we assume that the bucket is searched as a linked list. We considered index widths of 14 through 17 bits, corresponding to 2k to 128k hash buckets.

The hashing functions $f_{RAM,1}$ and $f_{RAM,2}$ are shown in Figure 9. Function $f_{RAM,1}$ is similar to the functions used for the CAM, whereas $f_{RAM,2}$ uses the standard IEEE 802.X 32-bit CRC function. The CRC function is used because it is a simple way to generate pseudo-random keys in hardware. Figures 11a-b plot the average number of comparisons per table lookup. We see that both functions $f_{RAM,1}$ and $f_{RAM,2}$ are effective and conclude that the simple byte manipulations of $f_{RAM,1}$ are sufficient for both a hardware or software implementation.

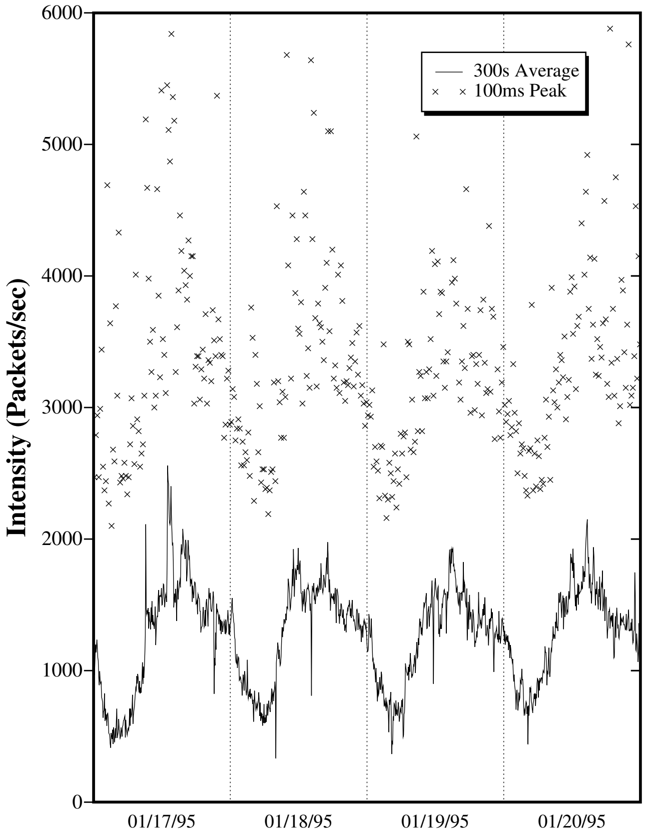


Fig. 12. Rate at which datagrams enter and leave the Berkeley campus.

F. Datagram Throughput Requirements

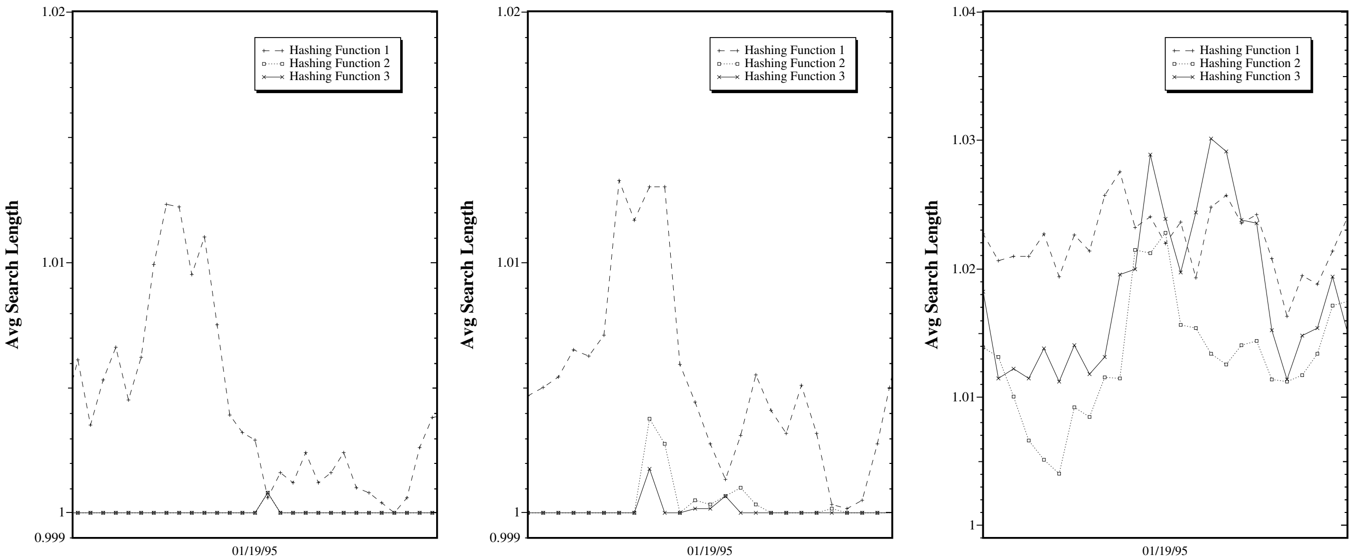
The BGW processes every datagram in a way significantly different from normal boundary routers. Specifically, the BGW must lookup the TCP connection record, update the byte and packet counters, and store the results back in the connection table. Figure 12 shows the peak and average datagram intensities. Our analysis showed that the BGW must be capable of processing datagrams at rates up to 6,000 packets per second. Depending on the architecture of the BGW, this seems to be quite feasible.

IV. BayBridge Prototype

In this section we show how the BayBridge, a high performance prototype router, could be used as the BGW for our billing system.

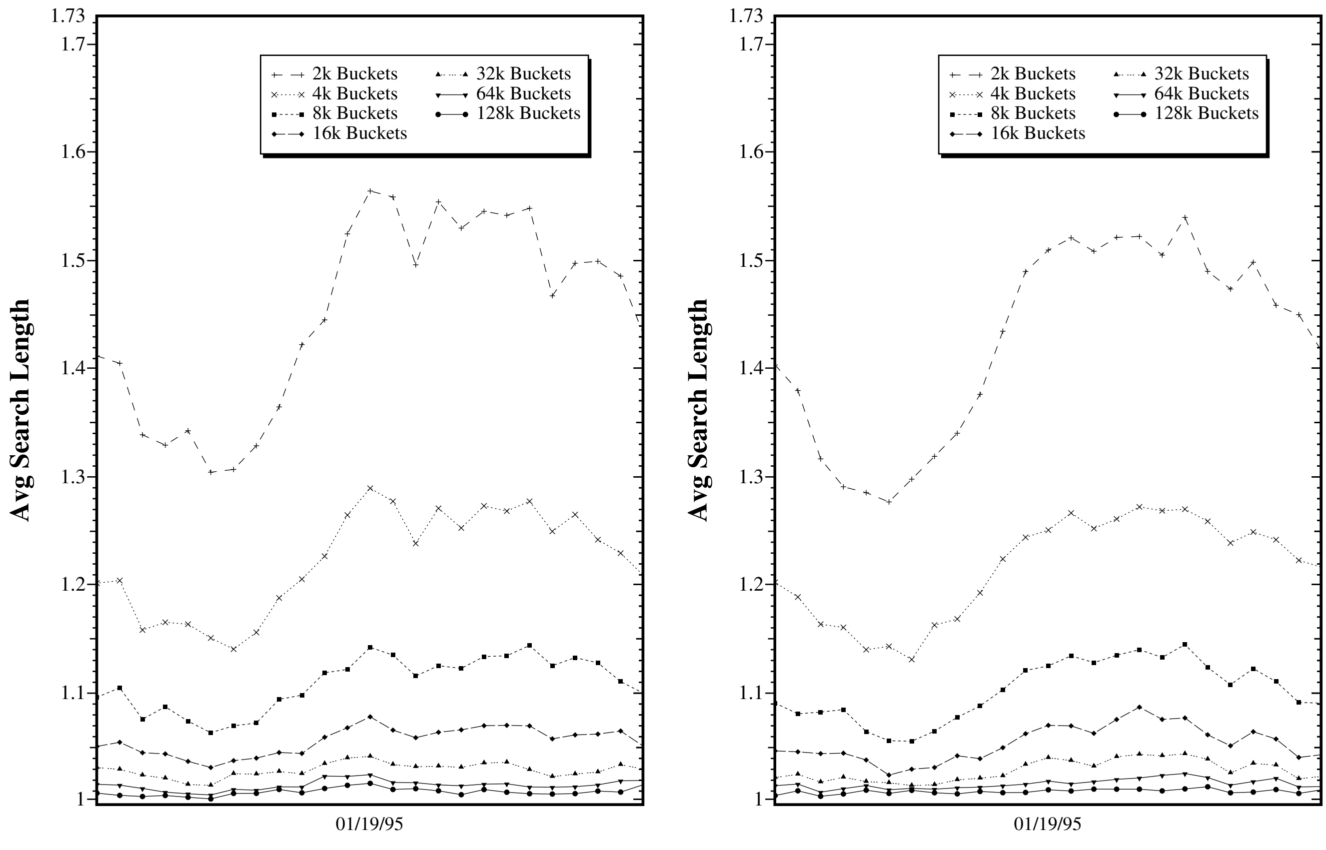
A. Overview of BayBridge

The BayBridge [13],[14] is a two-port bridge and router designed for high performance and flexibility. With both ports connected to FDDI rings, the BayBridge has been demonstrated to make routing decisions for over 200,000 datagrams per second. Referring to the system architecture in Figure 13, datagrams are processed at the port on which they arrive by a programmable protocol converter which consults a CAM-based address table. The datagram may be forwarded to the other port or sent to the local host. The address tables contain 1024 entries, each entry consists of



(a) 48-bit CAM (b) 32-bit CAM (c) 16-bit CAM

Fig. 10. Average number of lookups for each connection in CAM-based connection tables.



(a) $f_{RAM,1}$ (b) $f_{RAM,2}$

Fig. 11. Average number of lookups for each connection in RAM-based connection tables.

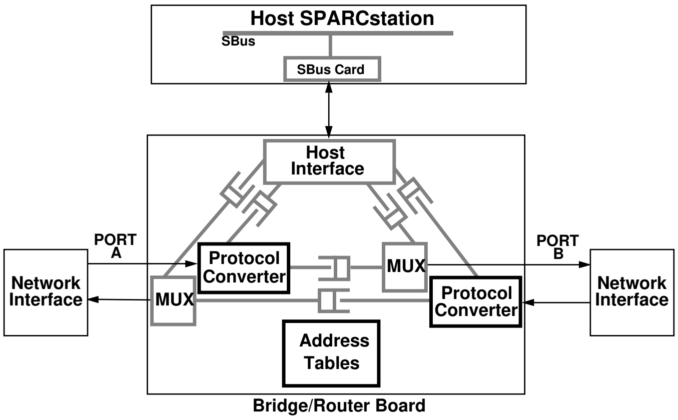


Fig. 13. The architecture of the BayBridge prototype.

a 48-bit wide CAM entry, and a shadow entry in RAM.

B. Using the BayBridge as a BGW

The BayBridge may be used as a BGW in the following way. The CAM-based address tables are implemented as described in Section III.E. The connection tables are shown in Figure 8: hashing function $f_{CAM,3}$ is applied to the 96-bit connection identifier to generate a 48-bit index which is used to find a match in the CAM. To verify the match, one of the IP addresses and one of the port numbers are maintained in the shadow RAM entry. The RAM entry also contains counters for bytes and packets in each direction. The next hop routing information is also contained in the shadow RAM entry.

When a TCP message arrives, the protocol converter searches in the connection table for a matching connection. If the entry is not found then this message must be from a new connection. The message is forwarded to the BayBridge's local host which forwards the connection identifier to the Access Controller. When the connection has been accepted, the host accesses the connection tables directly to insert the new entry.

If the entry is found in the connection table, the counter entries are read, incremented by the protocol converter and written back to the table.

C. Feasibility of BayBridge as a BGW

To operate as a BGW, the BayBridge must be able to handle the intensity of connection setups and have sufficient space in its connection table for all active connections. It must also be fast enough to process all datagrams as they arrive, finding the entries quickly in the connection tables.

We found in Section III.C that the intensity of new connections does not exceed 79 per second. This is easily achievable by the BayBridge: the protocol converter and host can forward many hundreds of packets per second to the Access Controller.

Section III.D showed that if a 60 minute inactivity timeout is used, the maximum number of active connections is 1,350. The BayBridge has a table size of just 1,024 entries: too small to hold all active connections. One way to overcome this limitation is to use the hardware CAM-based address tables as a cache for a larger connection table in the host's memory.

In Section III.F we showed that the maximum arrival rate for datagrams is less than 6,000 packets per second. Assuming that the first lookup is successful, the BayBridge can perform the metering and routing functions in approximately $13\mu s$[6]. This corresponds to a sustainable rate of 75,000 packets per second simultaneously in each direction.

We conclude that the BayBridge is easily capable of acting as a BGW between the Berkeley campus and the rest of the Internet.

V. Other Relevant Measurements

A. Variation in WAN Usage Levels

It is intuitive that different Internet hosts and users generate widely different amounts of traffic. This intuition is a crucial part of the motivation for usage based cost recovery. However, intuition alone should not justify complex cost recovery systems. We now show that there is a very large variation in WAN usage by hosts and by users on the Berkeley campus.

Table II shows the usage per-host averaged over the duration of the trace for the top-ten most active subdomains within Berkeley's network. These statistics indicate that the per host average usage varies widely by subdomain. The EECS Department subdomain has the largest number of hosts. The amount of WAN TCP usage by these hosts averaged 23 megabytes with a standard deviation of 56 megabytes. Figures 14a-b show the large variation in usage for these hosts.

Figures 15a-b show the distribution of usage per user.[7] For this set of users, the average amount of usage was 1.0 megabytes with a standard deviation of 3.7 megabytes.

B. Potential for Congestion Pricing

In addition to cost recovery, a billing system can also be used to control congestion by enforcing a pricing scheme. Using our trace data, we consider congestion control at two different time granularities and measure their effect on network traffic. These results are only intended to give an indication of the possible improvements in network efficiency. We do not propose specific techniques for achieving these improvements.

In the first scheme, we assume that the price to use the network varies by time of day. For example, traffic sent during the day may be more expensive than at night. Some traffic is not sensitive to time of delivery and may be postponed to a less expensive time of the day. In particular, we assume that under this scheme email and bulletin-board traffic[8] is "non-interactive" and therefore

[6] The published version of this paper contained a typographical error (ms instead of μs). We have corrected it here.

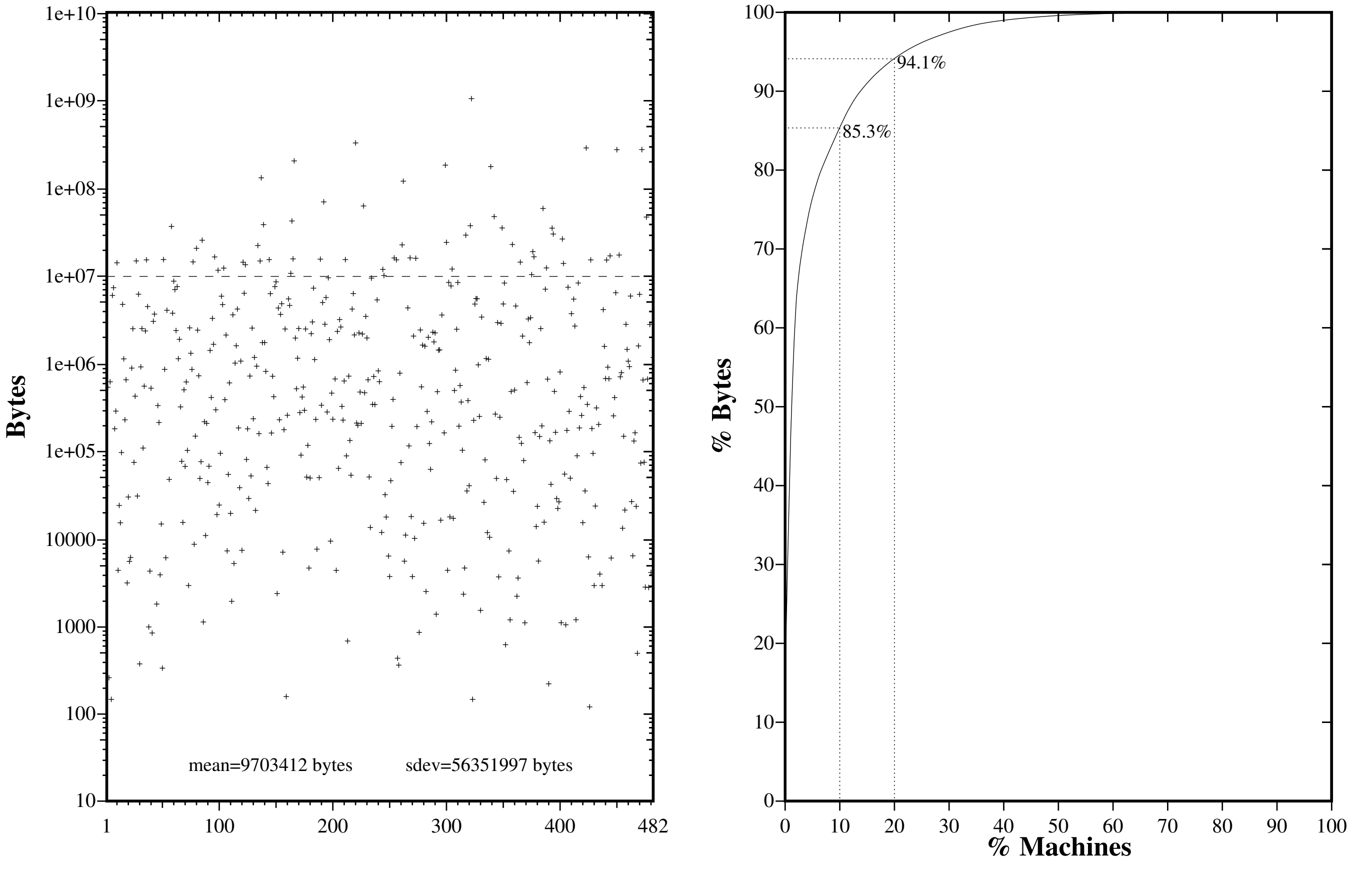
[7] This data is from an earlier trace recorded from midnight November 10 through midnight November 14, 1994.

[8] These are the smtp and nntp protocols. They accounted for 9% and 24%, respectively, of all wide area data bytes during our trace.

TABLE II
AVERAGE USAGE BY HOSTS IN TOP-TEN SUBDOMAINS.

Subdomain	Number of Hosts	Per Host Averages[a] Connections	Datagrams	Bytes	Percentage in Category[b] Datagrams	Bytes
cs.berkeley.edu	469	1,393	130,737	23,422,981	26.6%	26.9%
eecs.berkeley.edu	482	594	56,633	9,703,412	11.8%	11.5%
lib.berkeley.edu	395	480	41,645	7,620,778	7.1%	7.4%
math.berkeley.edu	115	677	108,682	24,225,511	5.4%	6.8%
biochem.berkeley.edu	111	803	95,472	23,567,586	4.6%	6.4%
hip.berkeley.edu	1,058	146	13,364	1,935,854	6.1%	5.0%
ssl.berkeley.edu	79	314	77,873	19,349,504	2.7%	3.7%
me.berkeley.edu	235	249	35,406	6,427,425	3.6%	3.7%
cchem.berkeley.edu	295	195	27,295	5,143,642	3.5%	3.7%
geo.berkeley.edu	56	1,656	129,543	26,164,881	3.1%	3.6%

[a] These are four day averages.
[b] This table and its percentage calculations exclude hosts that provide services to the entire campus.



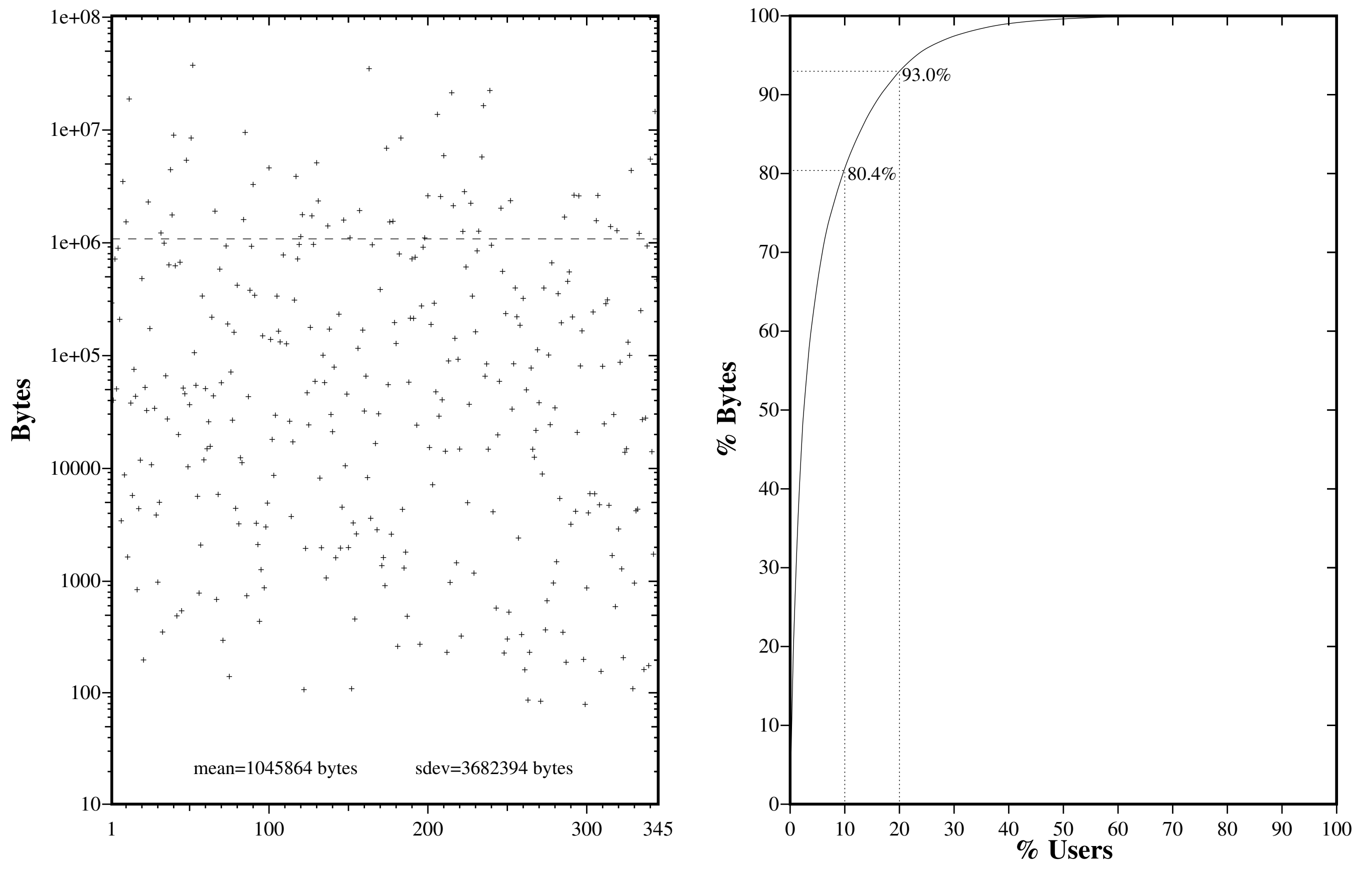
(a) Scatter Plot (b) Cumulative Distribution

Fig. 14. Variation of network usage by machines in the EECS.Berkeley.EDU subdoamin.

can be delayed. The remaining traffic is assumed to be "interactive."[9] We spread the non-interactive traffic over time by "water-pouring," i.e., we distribute this traffic so as to level-out the total (interactive + distributed non-interactive) intensity as much as possible.

[9] The most active of these "interactive" protocols were ftp, WWW, shell, telnet and "X." These accounted for 33%, 16%, 4%, 3% and 3%, respectively, of all wide area data bytes during our trace.

Figure 16 shows the improvement that can be achieved using "water-pouring." The top line shows the intensity of all network traffic in bytes per second. The lowest line shows the intensity of just the interactive traffic. The horizontal line shows the intensity if the non-interactive traffic is "poured" over the interactive traffic. By spreading the



(a) Scatter Plot (b) Cumulative Distribution

Fig. 15. Variation of network usage among a selected population of users.

non-interactive traffic in this way, the network utilization is almost constant at an intensity of 150,000 bytes per second with a peak of 170,000 bytes per second. This represents a 23% reduction in peak bandwidth.

In the second scheme, we assume that the network is controlled more rapidly in response to congestion. This may correspond to a flow-control mechanism, traffic-shaping using buffering or a policy in which the price varies quickly in response to congestion. To estimate the potential gain of such a scheme, Figure 17 shows the busiest 0.1 second and 1 second intervals during the trace, measured using a moving-average buffer (for comparison, the figure also shows the average). The peak intensity over 1 second is 22% below the peak intensity over 0.1 second, suggesting that the peak network load can be reduced dramatically by such a scheme using a buffer of just 250,000 bytes.

VI. Conclusion

More than 90 percent of an estimated $200 million annual Internet cost is today recovered from users in the form of a variety of organizational charges and user prices. These charges and prices are designed for administrative convenience rather than to promote efficient use of network resources. Prices designed to achieve efficient allocation or fair cost recovery, however, require a billing system that can meter the individual user's traffic and that can present the user with information that encourages efficient use. This paper presents a billing scheme that can identify and authenticate users, monitor individual user traffic, and present the user with real time pricing information. An implementation of the scheme based on the BayBridge, a prototype router, demonstrates that the billing scheme is practical, and introduces acceptable latency.

It is an open question whether the traffic demands and corresponding costs of operating the Internet will grow sufficiently to make pressing the need for an efficient pricing and cost recovery scheme. That will depend on the proliferation of bandwidth consuming applications on the Internet. In turn, that will depend upon the ability of Internet protocols, links and routers to support such applications in competition with other network service providers, including cable TV and telephone companies.

Statistics collected from a four day trace of all TCP traffic between the University of California, Berkeley, and the "outside world" show both homogeneities and diversities within the current University population. The data reveal several features that are relevant to the feasibility of different pricing and cost recovery schemes. First, if pricing schemes vary by time of day, then up to 33% of the traffic which represents e-mail and bulletin-board traffic can be

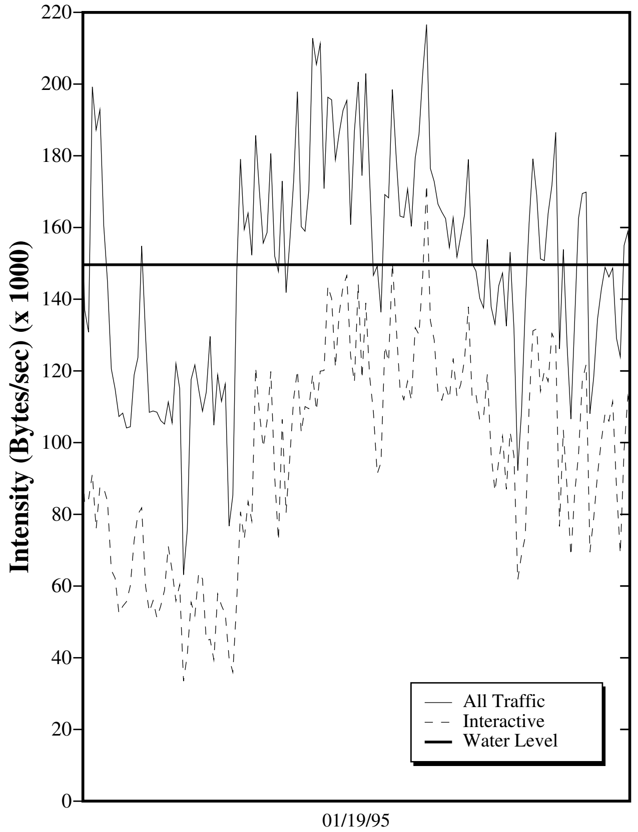


Fig. 16. Reduction in peak network bandwidth by "pouring" e-mail and bulletin-board traffic over the day.

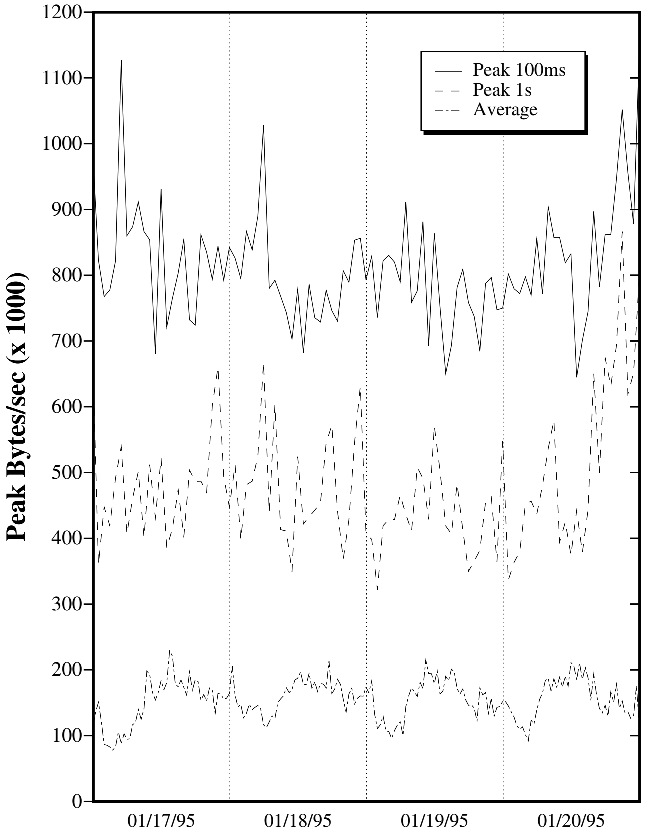


Fig. 17. The peak intensity, in bytes per second, of traffic leaving the Berkeley campus, measured over 0.1 second and 1 second intervals compared with the average intensity.

shifted over time with little loss in benefit to the user, since this traffic is not sensitive to time of delivery. Such a shift would reduce peak utilization. Second, on a much smaller time scale, we find that the peak intensity over 1 second is 22% below the peak utilization over 0.1 second. So if pricing schemes can shift traffic by one second, the peak router and link capacity can be significantly reduced. Such shifts may be automatically accommodated by increasing buffer sizes at hosts or in routers. Third, it seems feasible to recover cost for wide area traffic either from time of day pricing or from short term peak pricing. Recovery through time of day pricing would have a broader base and would be much simpler to implement. Recovery through short term peak pricing would be more difficult to implement, but it would also lead to greater reductions in the peak capacity of routers and links.

Acknowledgments

The authors appreciate the generous assistance of Cliff Frost, Kevin Mullally and Kevin Zimmerman in collection of network trace data.

References

[1] J. Mackie-Mason and H. Varian, "Economic FAQs About the Internet", *Journal of Economic Perspectives*, vol.8, no.3, pp. 75-96, Summer 1994.

[2] C. Mills, D. Hirsch and G. Ruth, "Internet Accounting: Background", *RFC 1272*, Internet Engineering Task Force, 1991.

[3] H.W. Braun, K.C. Claffy and G.C. Polyzos, "A Framework for Flow-Based Accounting on the Internet",*Proc. of the Singapore Intl Conference on Networks*, 1993.

[4] N. Brownlee, "New Zealand Experiences with Network Traffic Charging", *ConneXions*, The Interop Company, vol. 8, no. 11, November 1994.

[5] D. Estrin and L. Zhang, "Design Considerations for Usage Accounting and Feedback in Internetworks", *ACM Computer Communication Review*, vol. 20, no 5, pp. 56-66. October 1990.

[6] R. Braden, "Requirements for Internet Hosts – Communication Layers", *RFC 1122*, Internet Engineering Task Force, 1989.

[7] D. E. Comer, *Internetworking with TCP/IP — Volume I*, Prentice-Hall, 2nd Ed., 1991.

[8] J. Mackie-Mason and H. Varian, "Pricing the Internet", In *Public Access to the Internet* (Kahin, B. and Keller, J. eds.), Prentice-Hall, 1994.

[9] Taxpayer Assets Project - Information Policy Note May 7, 1994.

[10] K. Claffy, H.W. Braun and G. Polyzos, "Tracking Long-Term Growth of the NSFNET", *Communications of the ACM*, vol. 37, no. 8, pp. 34-45, Aug 1994.

[11] J. Steiner, C. Neuman and J. Schiller, "Kerberos: An Authentication Service for Open Network Systems", *Proc. of USENIX Winter Conference*, pp. 191-202, Feb 1988.

[12] Merit Network, Inc., "NSFNET Backbone Service Traffic Distribution by Internet Network Number," ftp://nic.merit.edu/nsfnet/statistics/1994/nsf-9409.pnets. September 1994.

[13] N. McKeown, R. Edell and M.T. Le, "The BayBridge: a high speed bridge/router". (Protocols for High-Speed Networks III. IFIP WG6.1/WG6.4 Third International Workshop, Stockholm, Sweden, 13-15 May 1992). *IFIP Transactions C (Communication Systems)*, vol.C-9:203-18, 1993.

[14] N. McKeown, R. Edell and M.T. Le, "The BayBridge: A High Speed Bridge/Router between FDDI and SMDS Part I — System Architecture and Performance", Submitted.

Richard Edell (S'94) received the B.S. and M.S. degrees in electrical engineering from the University of California, Berkeley, in 1991 and 1994, respectively.

He is currently working toward the Ph.D. degree in electrical engineering at the University of California, Berkeley. His research interests include network interface architectures, Internet economics and technology policy.

Nick McKeown (S'90-M'95) received the B.Eng degree in Electronic Engineering from the University of Leeds, England in 1986, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1992 and 1995, respectively.

He is currently an Assistant Professor of Electrical Engineering and Computer Science at Stanford University. His research interests include high performance ATM switching and economics of the Internet.

Pravin Varaiya (M'68–SM'78–F'80) received the Ph.D. degree in electrical engineering from the University of California, Berkeley.

He is James Fife Professor of Electrical Engineering and Computer Sciences at the University of California, Berkeley, and Director of California PATH, a multi-university program of research in Intelligent Transportation Systems. From 1975 to 1992 he also was Professor of Economics at Berkeley. He has taught at MIT and the Federal University of Rio de Janeiro. He was a member of the technical staff at Bell Laboratories during 1962-63. Dr. Varaiya has held a Guggenheim Fellowship and a Miller Research Professorship. His areas of research are communication networks, transportation systems, and electric power systems. He is writing a book with Jean Walrand on high-speed, flexible communication networks.