A RESOURCE BASED PRICING POLICY FOR REAL-TIME CHANNELS IN A PACKET-SWITCHING NETWORK.

Colin Parris and Domenico Ferrari
The Tenet Group
Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California
and International Computer Science Institute
Berkeley, California, U.S.A.

Abstract

In the packet switching networks of the future the need for guaranteed performance on a wide variety of traffic characteristics will be of paramount importance. The generation of revenue, to recover costs and provide profit, and the multiple type of services offered will require that new pricing policies be implemented.

This paper presents a resource based pricing policy for real-time channels (ie., channels with guaranteed performance) in a packet switching network. The policy is based on a set of specific criteria, and the charges for any channel are based on the resources reserved for use by the channel. This reservation charge is based on the type of service requested, the time of day during which the channel exists, and the lifetime of the channel. We argue that the traditional resources are not sufficient to determine a fair reservation charge for a channel offering guaranteed delay bounds, and we introduce the notion of a delay resource in our charging formula. The type of service requested is thus characterized by the amount of the bandwidth, buffer space, CPU, and delay resources reserved. The analysis of this pricing policy is reduced to the analysis of a single node of the network, assuming a homogeneous network. This single-node characteristic increases the scalability and flexibility of the policy. An example of an implementation of this policy is provided.

This research was supported by the National Science Foundation and the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives, by AT&T Bell Laboratories, Hitachi, Ltd., Hitachi America, Ltd., Pacific Bell, the University of California under a MICRO grant, and the International Computer Science Institute. The views and conclusions in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of any of the sponsoring organizations.

1. Introduction.

With the ever increasing desire for multimedia applications on computer networks, the need for real-time communication services (i.e. services with guaranteed performance) is readily apparent. In conjunction with the fulfillment of this need, efforts are now being made to determine pricing scheme formats suitable for use in this type of networking environment. While pricing policies exist for non real-time networks (datagram networks) as well as for conventional public utilities networks (telephone and power system networks), they cannot fulfill the requirements that we desire of a pricing policy for an integrated-services network.

This paper presents these requirements, discusses the inadequacies of the conventional pricing policies, and proposes a new pricing policy. The paper begins in Section 2, with the requirements of a useful pricing policy and an overview of the disadvantages of conventional schemes. Next, Section 3 presents and analyzes the proposed pricing scheme. An example of the scheme is discussed in Section 4. Section 5 concludes the paper.

2. Motivation for the Pricing Policy.

This section states the requirements of a useful network pricing policy, provides a brief overview of the conventional pricing schemes, and discusses the disadvantages of each scheme.

The requirements of a network pricing policy are:

- Allow the service provider to collect revenue commensurate with the quality of service provided to the client.
- Allow clients to predetermine the charges for the services they need.
- Discourage client actions that will decrease the efficiency of the network.
- Allow the policy mechanism to be implemented with minimal overhead.
- Allow a simple relationship between the performance characteristics of the network and the revenue derived from the network.

Although most of the requirements stated above are self-explanatory, a brief discussion of some of these requirements may provide some further insight. It is essential that the clients be able to predict and verify the charges for their usage. In some pricing schemes the price of the resource is based on the demand at that time, and clients can bid for the resources that they need. The demand can be viewed as a function of the state of the network and can not be predicted easily. Thus, with such a scheme, clients would not be able to forecast their charges.

The pricing of network services should encourage clients to undertake actions that increase the efficiency of the network and discourage actions that decrease the efficiency of the network. This efficiency should be thought of as a combination of performance and pareto efficiency. A relevant example exists in the use of video conferencing. Video conferencing sessions should generally take place during "working hours", as most of the members involved in the session would generally be available during those hours. Writing large video files to multiple archive video servers could possibly utilize resources equivalent to those of video conferencing, thereby maintaining the same level of efficiency in the network. However, the conferencing session should be given preference over the archive session as by doing so the pareto efficiency is increased. This preference could be given by charging a higher price for service during a specific time of day, thus forcing "time of day" insensitive traffic out of "working hours". The clients sending the videos to be archived are likely to be discouraged, by the higher charge, from network usage at specific intervals during the day.

The relationship between the performance characteristics and the revenue generated should be such that the effect on the revenue of any change in network resources can be easily determined. This is essential in dealing with the loss of resources due to failures in the network as well as in the quick prediction of the additional revenue that would be generated by the buying or leasing of additional resources or the addition of new guaranteed performance services.

2.1. Disadvantages of Conventional Pricing Policies.

The conventional pricing policies that will be discussed here are: Flat Per-Packet pricing, Peak Load pricing, and Priority Based pricing. A brief overview of each policy will be provided and its inadequacies, with respect to the requirements presented above, stated. A more thorough treatment of each policy can be found in the corresponding reference.

In the Flat Per-Packet pricing policy users are charged according to the number of packets that are sent by them to any destination. This policy fails to satisfy two of the requirements, ie., (1) and (3), stated above. The revenue collected by the service provider is not commensurate with the service provided in that a client who uses a real-time service and sends p packets and a client who sends a non real-time message with p packets are charged the same amount. In effect, the non real-time users are carrying some of the real-time users charges. Clients are not discouraged from inefficient actions. As the charge to send p packets is the same regardless of time of day or network state, clients are not discouraged from sending low priority data during peak hours.

Peak Load pricing policies assume that there are regular times of the day at which the load on the network is heavy. The charge for usage is raised to shift flexible clients off the peak load intervals. When the peak periods are predictable (Static Peak Load pricing), the clients are made aware of both the period and the charges for the period, at some predetermined time before the peak period. When the peak periods are not predictable (Dynamic Peak load pricing), peak load rates and their corresponding charges are varied dynamically with the network load. The charges are applied to the duration of time the channel existed or the number of packets sent during the peak load and non peak load periods [1]. In the case of Static Peak Load pricing, requirements (1) and (3) are not met. The first three requirements, (1),(2), and (3), are not met in the case of Dynamic Peak Load pricing. As with the previous policy, all the clients who use the service during the peak periods must receive the same quality of service, otherwise the clients with lower quality service will be paying for the clients with higher quality service. The same can be said for the non peak periods. In the case where clients are charged based on the number of packets sent, clients may acquire connections and not send any traffic on them, thereby blocking efficient clients from using them while not allowing the service provider to collect any revenue on these connections. In the case that the periods are not predictable, the clients are not able to predetermine their charges since the charges are changing dynamically. Another problem with the pricing mechanism is that the clients cannot verify their charges, as they must rely on the network to give them the current price per packet, which may vary dynamically. In Static Peak Load pricing, clients know the number of packets sent and can verify their charges without network intervention. While Static Peak Load does discourage some inefficient actions by the clients by forcing flexible clients off peak periods, it does not deal with inefficient actions in non-peak periods, nor does it deal with inefficiencies among clients who remain in the peak periods. Further granularity is needed in discouraging such actions.

The last policy to be reviewed here is that of Priority Based pricing. In this policy, the network will serve clients in the order of their priority levels, and the per-packet charges will be computed accordingly. This policy is more adaptable than Peak Load pricing, as the priorities present a basis for the network to delay lower priority traffic in favor of higher priority traffic. The clients are allowed to set a certain priority level for their traffic , and, if the experienced performance is too low, they may increase their priority level until their performance criterion is met. The per-packet rates may be set and changed due to the demands of the other clients as they increase their priority levels [2]. The main problem with this scheme is that it fails to meet requirements (1), (2) and (4). As with the Peak Load pricing policy, the service provider may not collect revenue commensurate with the service provided, as clients may acquire connections and not use them. This inhibits the use of the connection by other clients, resulting in loss of revenue to the network, as clients are charged on the basis of the packets sent, not on the duration that their connection is held. Clients cannot do productive long-term budgeting as the charges vary dynamically, and the problem of charge verification exists also in this case. Implementation is difficult in this scheme since each packet must be examined to determine its priority and the accounting mechanism will have to record the packets, their priorities, and the associated costs. The bidding among clients as they vary their priorities will also have to be regulated by the accounting mechanism. The overhead of this policy may be unbearable.

A simpler version of priority-based charging is one in which the priority of packets is based entirely on the quality of service that the clients select and is fixed. There is no bidding to increase priorities, as these priorities can only be increased by acquiring a higher quality of service. The charges for each quality of service are fixed and available to the client. Charges are per-packet, and are applied to each packet dependent on its level of service. This policy satisfies requirements (2), (4), and (5). Requirement (1) and (3) are not satisfied, as clients can acquire connections and not send any packets across them. While these clients are not using any of the resources allocated to them, they are decreasing the efficiency of the network by not allowing other clients access to those resources. The total revenue of the network is also decreased, as no revenue is generated by these connections. This policy will be referred to in the sequel as the Multiple Fixed-Priority Based Policy. An example of this pricing policy is given in [3].

As can be seen from the reviews presented above, the conventional schemes are inadequate for our purposes. A new policy is needed that meets all the requirements stated above. This policy is presented in the next section.

3. A Pricing Policy for Real-Time Channel Establishment Scheme.

In this section, the proposed pricing policy will be presented. It will then be analyzed with regard to the requirements, and applied to our real-time channel establishment scheme. The reader is assumed to have some knowledge of the traffic characterization and the channel establishment scheme found in reference [4]. For the sake of simplicity, we assume that only real-time traffic will be present on the network. It should be noted however that the policy could easily support non real-time traffic services.

3.1. The Proposed Pricing Policy.

In its simplest format the proposed pricing policy may be expressed as:

$$Total\_Charge_i = Reservation\_Charge_i \ ,$$

where $Reservation\_Charge_i$ is the charge applied for the reservation of resources for channel *i*. All of the resources reserved for the channel are available for immediate utilization by that channel.

The equation given above can also be written as

$$Total\_Charge_i = TOS_i \cdot t_i \cdot tod_i \ ,$$

where

- TOS_i is the type of service required by the client for channel *i*.
- t_i is the lifetime of channel *i*.
- tod_i is the period factor (time of day) associated with channel *i*.

TOS_i is chosen from a menu specifying the performance parameters of the service provided. There are *n* different *TOS* and TOS_i is the service type from 1..*n* requested by channel *i*. Each of the *n* *TOS* has a fixed charge per unit time associated with it, and these prices are made available to all clients. tod_i is the factor associated with peak periods. This factor is used to encourage clients with flexible needs to shift their workloads to periods where the *tod* factor is less. The tod_i values for each period are available to the client. The value of *tod* associated with channel *i* changes if the channel is present when a period ends and another begins, and the effects of such changes in the charge are calculated during accounting. t_i is the duration of the channel. $TOS_i \cdot t_i$ gives the charge due to reserving the resources for the service type given by TOS_i for time t_i . The factor tod_i influences this charge based on the periods during which the resources were reserved.

To be adequate, this pricing policy needs to meet all of the requirements given above. The policy satisfies requirement (1), since clients with higher performance service requests get charged a higher rate. The service provider can scale the value of TOS_i to recover costs and realize a reasonable profit. The reservation of resources guarantees that these resources are available to the clients when they demand them, but the resources may be utilized by other clients (usually the non real-time clients) when not in use by their "owners". If the client's request is accepted by the network, the reserved service is immediately provided.

Requirement (2) is met in that the TOS_i, and tod_i charges are fixed and available to the clients. Clients also know the pricing policy, and usually the lifetime of each channel (in some cases exactly, in some others only approximately), and can therefore determine the charges that their channels will incur, as well as verify the charges applied to those channels by the network.

Clients are encouraged to do efficient actions by the TOS_i and tod_i values. Clients needing less stringent levels of performance will choose a *TOS* with a correspondingly lower charge (i.e., a lower price is paid for a lower quality of service) and will have a reduced amount of resources available to them (as compared to clients with higher valued *TOS*). The use of tod_i will force flexible clients with the same *TOS* to shift loads off of the peak periods. If clients use the network inefficiently by reserving resources that they do not intend to use, they are penalized by having to pay the higher charges associated with that quality of service. This encourages efficient use by the clients and thus satisfies requirement (3).

Requirement (4) is met, in that the policy can be implemented with minimal overhead. The TOS_i value is determined at the time of establishment. Timestamps are kept upon the establishment and termination of all channels. The t_i and tod_i values are determined using these timestamps. The charges can then be computed offline and submitted at a later date to the client.

The last requirement to be met is that of scalability and flexibility. The revenue (i.e., the total charge) should be related to the network utilization so that changes to the topology or the service menus can be easily incorporated into the current pricing scheme. The scalability aspect refers to the addition of new nodes, while the flexibility refers to the addition of new services. Our policy meets this requirement, as pricing is handled on a per-node basis, and the total charge is the sum of the charges incurred in the nodes along the route. To examine the effects of additional real-time services in the network, only a single node need be analyzed. Therefore, network analysis actually collapses to the analysis of a single node of the network. Obviously, a homogeneous network is assumed in the analysis. This collapse of the analysis to a single node is accomplished by deriving the *TOS* value directly from a network utilization function. In the network utilization function selected, the bandwidth, cpu, buffer space, and delay resources are parameters. The function operates on these parameters and generates suitable *TOS* values. This utilization function and its relationship to the policy are discussed in the next section.

3.2. The Utilization Function.

In this section a description of the network and traffic models that were used in our simulations will be presented. A discussion will follow of the functions used to calculate bandwidth, buffer space, cpu time, and delay resource reservations.

In our network model, the network consists of an interconnection of nodes. Nodes are connected to each other using links. Nodes can be hosts or switches. In each case the classification as a node is based on the fact that the entity in question provides bandwidth, buffer space, and CPU cycles as resources for channels.

The traffic model and the performance specifications we used consist of the parameters as given below. A complete description of this model is available in [6]. The traffic and performance parameters are the following:

- $X_{\min}$ is the minimum packet interarrival time on the channel.
- X_{ave} is the minimum value of the average packet interarrival time over an interval of duration I.
- $S_{\max}$ is the maximum packet size in bytes.
- $T_{\max}$ is the maximum service time in the node for the packets.
- D is the source-to-destination delay bound for the packets.
- Z is the minimum probability that the delay of a packet is smaller than the given bound D.
- $W_{\min}$ is the minimum packet loss probability due to the buffer overflow.

The last three of these parameters define the quality of service offered by the network. The *TOS* value assigned to each channel type is based on the resources reserved by the channel and on the channel's effect on the admission of other channels into the network. The resources reserved are directly related to the client requirements, as can be seen in the equations below. If all nodes are identical, it is sufficient that these resources be determined for a single node for a complete analysis of the network.

$$\text{Bandwidth reserved} = \sum(1/X_{min)} \cdot S_{\max}$$

$$\text{Buffer space reserved} = \sum Buffers\,() \cdot S_{\max}$$

$$\text{CPU time reserved} = \sum(1/X_{min)} \cdot T_{\max}$$

$$\text{Delay resource reserved} = \sum(1/D) \cdot Z \cdot K\,(D,Z)$$

where:

- N_p is the number of nodes to be traversed by the channel.
- Buffers() is the buffer space function that calculates the number of buffers required by the channel at each node based on the values of client parameters.
- *K(D,Z)* is a heuristically determined delay function dependent on the parameters *D* and *Z.*

The bandwidth reserved in a node is determined by multiplying the maximum packet sending rate, which is the reciprocal of the minimum packet interarrival time, by the maximum packet size. The total bandwidth reserved is the summation across all the nodes in the path. If the nodes are identical, this summation is achieved by simply multiplying tge bandwidth reserved in a node by the total number of nodes in the path.

The total amount buffer space is determined by summing the result of the Buffers () function across all nodes multiplied by $S_{\max}$. A thorough description of the Buffers () function will be provided in Section 4.2 As explained above, this can be easily achieved by multiplying by the number of nodes in the path.

The *TOS* value is computed as follows:

$$\text{TOS} = \alpha \cdot \text{ Bandwidth reserved} + \beta \cdot \text{ Buffers reserved} + \\ \gamma \cdot \text{ CPU time reserved} + \delta \cdot \text{ Delay}$$

The values of the coefficients α, β, γ, and δ are chosen by the network designers to reflect the relative scarcity of each of the resources. For instance, if bandwidth is the bottleneck resource, α must be made larger than the other constants in order to encourage clients to use the resource effectively.

The bandwidth, buffer space, and CPU time resources are easily defined and are directly related to the quality of service requirements. However, the delay resource cannot be directly related to that quality. The delay resource can be thought of as a quantification of the loss of potential for the network to admit other channels due to the admission of the channel being created. The admission of a channel into the network may reduce the network’s ability to admit other channels due to the increased difficulty the node scheduler encounters in satisifying all the local deadlines. We therefore say that the new channel consumes some of the delay resource of the network. The delay resource is related to the delay (*D*) and the probability of no lateness (*Z*) requirements of the channel. We make the delay resource inversely proportional to the delay parameter, since the shorter the delay required by the channel, the greater the restrictions placed on the scheduler, and directly proportional to the probability parameter since the greater this probability, the greater the restrictions placed on the scheduler. The proportionality constant *K(D,Z)* is explained in the next section.

3.2.1. The delay function *K(D,Z)*.

The purpose of the delay function , *K(D,Z)*, is to bias the values of the *D* and *Z* parameters of all the different channel types so that the amount of the delay resource consumed by each channel of a specific type correctly reflects the loss of potential to admit other channels into the network.

Before an explanation of *K(D,Z)* can be provided, a few terms need to be defined. The *request space* is an *n*-dimensional region containing all possible combinations of the *n* types of channels which can be requested by clients. In the example to be provided in section 4, there are three channel types, thus the *request space* is a 3-dimensional space. The values of each of the 3 elements of a vector in the *request space* are non-negative, and