Exploring Traffic Pricing for the Virtual Private Network

Zhangxi Lin
College of Business Administration
Texas Tech University
zlin@ba.ttu.edu

Peng Si Ow
Center for Research in Electronic Commerce
The University of Texas at Austin
phighnam@aol.com

Dale O. Stahl
Department of Economics
The University of Texas at Austin
stahl@eco.utexas.edu

Andrew B. Whinston
Center for Research in Electronic Commerce
The University of Texas at Austin
abw@uts.cc.utexas.edu

June 25, 2001

Exploring Traffic Pricing for the Virtual Private Network[1]

Abstract:

This paper explores the implementation issues of network traffic pricing in Internet-based virtual private networks (VPNs). A transaction-level pricing architecture based on proxy server technology is proposed. A simplified VPN traffic-pricing formula is derived for optimizing VPN bandwidth service welfare. We provide price formulae for both prioritized first-in-first-out bandwidth scheduling and non-prioritized round-robin bandwidth scheduling. A prototype traffic-pricing system, *VPN Traffic-Pricing Experiment System* (VTPES), has been developed to test the transaction-level pricing architecture and examine the pricing formula. Experiments conducted with VTPES show that the pricing mechanism can effectively improve a VPN's transmission efficiency.

[1] This research has been sponsored by Intel Corp., Lucent Technologies Inc., and Mannesman Ag.

1. Introduction

The virtual private network (VPN) [15] is a value-added network built upon all types of network clouds, particularly on the Internet, with secured virtual paths for data transmission. The explosive growth of the Internet is also spilling over to Internet-based VPNs, so that VPNs are emerging as an important enterprise networking solution for corporations. There are numerous VPN products on the market now, such as: e-Network by IBM, MultiVPN by Ascend, VTCP/Secure by InfoExpress, SmartGate by V-ONE, and VPN-1 by Check Point. Forrester Research Inc. [http://www.forrester.com/], an industry analyst, revealed that the cost for today's Internet-based VPN for 1,000 users is one-third of the cost of a traditional private network with dedicated lines of the same capacity. This cost effectiveness has translated into a fast growth of VPNs. According to Infonetics Research, worldwide expenditures on VPNs were $205 million in 1997, and should double annually through 2001 when they are expected to reach $11.9 billion. Although the Internet-based VPN has brought users cost-effective solutions to support network applications, Internet traffic is becoming more congested as a result of exponentially increasing traffic loads. These loads substantially diminish the net benefits of users and service providers. In its "*Top 10 Discoveries About the Internet*" [http://www.keynote.com/measures/top10.html], Keynote Systems listed the Internet performance problem as the first one among the ten. Keynote System's *Keynote Business 40 Internet Performance Index* showed that the best response time was about 1.5 seconds in a web site access and the worst average was about 15 seconds.

Since the Internet came into use, computer scientists have made great efforts to improve the performance of the Internet and have successfully protected the Internet from collapses that happened earlier due to congestion. This is owing to a group of flow-control algorithms such as *slow-start* and *congestion avoidance* proposed in the late 1980s and widely implemented today [12][21][29]. In addition, queue management algorithms for Internet transmission nodes have been designed to complementarily allocate bandwidth and deal with queue overflow for network nodes along data paths [3]. Recent research in active queue management algorithms, such as *Random Early Detection (RED)* [8] and *fair queuing (FQ)* [5], has led to more powerful networking products.

The major idea behind these technical schemes is feedback. With feedback in a flow control session, the sender and the receiver computers of an end-to-end connection dynamically exchange the information of the available bandwidth and maintain an appropriate transmission bandwidth. Furthermore, heavily loaded nodes can send back alerts to the origins of data flows to trigger responses to the congestion. If the network communication software for the sender is responsive to these mechanisms, it will automatically reduce the data rate to avoid the congestion.

However, there are two problems with these non-incentive-compatible approaches. First, they are effective only if the network applications are responsive. Those data flows generated by "non-responsive" applications can get around the flow-control mechanism to obtain more bandwidth [7], and therefore deteriorate the network performance. Because of the ever-increasing heterogeneity of Internet protocols [31], many of which do not comply with the traffic control algorithms, these traditional approaches are no longer working properly. Second, a pure technical scheme is unable to discriminate among different types of data flows in accordance with their values. The only constraint to the overexploitation of the network bandwidth resource is the throughput time when the network is overloaded. In this case, the data flows sensitive to the delay are affected more regardless of their value to users. Presently, there is no final solution for the Internet congestion problem. Hence traffic congestion control is still an important research topic [1][6].

Since the 1990's, there is an emerging consensus that the Internet traffic congestion problem is not merely an engineering issue, but a problem of allocating scarce network resources to users whose valuations of these resources vary. Increasingly, research has been conducted into economic network resource allocation mechanisms that support usage-based pricing and incentive compatibility [4][11]. Examples are *dynamic bidding for access* by MacKie-Mason and Varian [20]; *priority pricing* by Gupta, Stahl and Whinston (denoted as the GSW model) [9][10][17]; *edge pricing* by Shenker, Clark, Estrin and Herzog [27]; *Paris metro pricing* by Odlyzko [22]; and *progressive second price auction* by Lazar and Semret [16].

Recently, *differentiated services* (DiffServ) [2] is emerging as a promising architecture for network traffic pricing. With the type-of-service (TOS) field in the IPv4 header, DiffServ allows for the prioritized transmission service. *Dynamic capacity contracting* by Kalyanaraman, Ravichandran and Norsworthy [13] is one of the efforts in this direction. Economic approaches to solving the network congestion problem are incentive-compatible in that they introduce the price, an extra feedback from network-traffic conditions to data flows to control their rate discriminatingly, so they can provide the mechanism to optimize network service benefits.

Although researchers theoretically proved that the economic approach has the potential to solve Internet congestion problems, they also realized that a good traffic-pricing model must come with an implementation scheme that is proved to be practical. It must be convincing to computer scientists that a mathematically intensified economic approach would not disturb the operation of the current mechanism for network traffic management and congestion control, and would work well alongside the existing technologies. Issues in the traffic-pricing implementation system include:

- how to build a traffic-pricing model fitting packet-switched networks that are playing a key role in the Internet;
- how the pricing mechanisms adapt to the network flow-control mechanisms and congestion-management schemes, and jointly contribute to the quality of service; and
- how to design a user-acceptable interface for network traffic pricing.

Our research in VPN traffic pricing explores the implementation feasibility of the network traffic-pricing scheme, and aims at potential demands from industry. We choose the Internet-based VPN as the target network according to the following considerations:

- The VPN's performance has become a significant issue in its business applications because the encrypted packets impose more traffic loads into the Internet, which has been eliminating the benefits from the VPN.
- The VPN possesses some useful business features such as user authentication and user account management that make the implementation of traffic-pricing feasible. In another aspect, a public network does not provide user account management functions, so traffic pricing would be practically very difficult.

We follow the methodology used by the GSW model, which was initialized by Stahl and Whinston in 1991 [28] and enriched by Gupta, Stahl and Whinston in 1996 [10]. The insights embedded in the GSW model include:

1) the existence of a stochastic equilibrium of the Internet economy with traffic pricing, in which a unique bandwidth resource allocation and an optimal dynamic bandwidth price maximize the overall service benefits; and

2) the fact that this stochastic equilibrium can be reached with a decentralized pricing system which dynamically adjusts the prices according to local traffic status.

The GSW model is a general equilibrium model with a resource-price structure that is incentive compatible for network resource allocation. The model was tested under various scenarios by a simulation of a public network. The simulation indicated that traffic pricing can significantly improve network service benefits and the service prioritization will lead to better outcomes. Encouraged by the GSW model's results, we have moved on to focus on developing a prototype VPN traffic-pricing system architecture and technology, and address issues related to the implementation of a practical network traffic-pricing system.

This paper is organized as follows: In Section 2 we initiate our discussion by proposing a transaction-level implementation architecture for a VPN traffic-pricing system. In Section 3 after charactering the bandwidth service of the Internet-based VPN, we derive a dynamic traffic-pricing model for the VPN. It is a customized GSW model for the VPN with the extension in non-prioritized round-robin scheduling. Section 4 introduces a prototype traffic-pricing system called *VPN Traffic-Pricing Experiment System* (VTPES), an attempt to bridge the gap between the theory and the practice in network traffic-pricing research. It is different from the software-based simulation system for the GSW model in that it is built on a configurable network composed of several computers and it uses real-time data flows for experiments. A number of experiments presented in the section show the effectiveness of the traffic-pricing scheme.

2. Transaction-level Traffic-Pricing for the VPN

A typical Internet-based VPN can be built up with special network devices to connect geographically distributed LANs into a virtual intranet/extranet over the Internet. The constructive hardware includes firewalls, certificate authority servers, security gateways, etc., underpinned by the security technologies such as security transmission protocols (e.g. IPSec), user-authentication information management protocols (e.g. LDAP), key-management protocols (e.g. ISAKMP), etc. Logically, we refer to the hardware set supporting the VPN technology as the *VPN gateway*. Therefore, a VPN gateway is an enhanced network gateway between an application domain and the Internet. It can provide the required security functions such as:

- Wide-area network tunneling, i.e., establishing a secure network connection across the public Internet;
- Data encryption;
- Filtering/firewalling, i.e., security control of incoming and outgoing packets at network edges; and
- User authentication.

Although the VPN is a cost-effective solution for enterprise networking, the application of encryption/decryption and related security protocols has added more traffic and processing overhead to the network. This has been blamed for the worsened congestion problem in the VPN. Here, we define network traffic congestion as prolonged time for a data flow to get through the network compared with the time it would take to get through an idle network[2]. By this definition, the congestion may result from the queue waiting time at network nodes as well as from queue overflow. The objective of VPN traffic management is to suppress lower-valued data flows through the Internet connections and maintain the overall network traffic at a certain level. In this way, congestion will be under control and the network service welfare can be

[2] A typical technical explanation of the cause of congestion is queue overflow that results in packet tosses and hence incurs retransmissions. The consequence is that the situation gets worse and transmissions are overwhelmingly delayed.

improved. The act of dynamically choosing bandwidth rental prices according to the network traffic status is the key to eventually optimizing the total benefits of VPN transmission services.

We propose a *transaction-level pricing* architecture for the VPN traffic pricing to solve implementability and efficiency problems. We claim that it allows us to solve relevant issues for a network traffic-pricing system, such as: digital contracting; pricing system efficiency; logistic system operation (payments, accounting, etc.) irregularities; problems involved with the integration of the traffic-pricing system and existing traffic-control techniques; and user acceptability.

Transaction[3] is a synonym for *job*[4], which is defined as a unit of a network service requested by a user. A job may generate one or several data flows transmitted through a network, where a *data flow* is a group of IP packets controlled by, for example, a TCP connection. We use *job size* to refer to the total volume of data flows incurred by a job. Job size is measured in the number of segments composed of several packets in the same data flow. In default, we will use *segment* as the unit of data flows. We use *throughput time* to describe the transmission time for a group of packets, e.g. a data flow, going through a section of transmission channel. The throughput time of a data flow includes its transmission time and waiting time at a given channel. It is determined by the traffic load conditions as well as routing and transmitting disciplines. Transaction-level pricing is to be implemented on top of transport layer protocols without looking into the internal mechanism at the lower layer for data flow delivery and control. With the transaction-level implementation, VPN gateways can schedule data transmission tasks in regard to the application needs and priorities.

---

[3] The definition of *transaction* by whatis.com (http://www.whatis.com): "In computer programming, a transaction usually means a sequence of information exchange and related work (such as database updating) that is treated as a unit for the purposes of satisfying a request and for ensuring database integrity. For a transaction to be completed and database changes to made permanent, a transaction has to be completed in its entirety."

Proxy server-based VPN traffic pricing is the underpinning infrastructure for transaction-level VPN pricing architecture. A proxy server is employed as the bandwidth broker to schedule the data flows with a pricing mechanism for an affiliated VPN gateway, which is called *traffic proxy server* (TPS). TPSs can be deployed somewhere between VPN security devices and application domains, for example, between VPN gateways and local area networks (Figure 1).

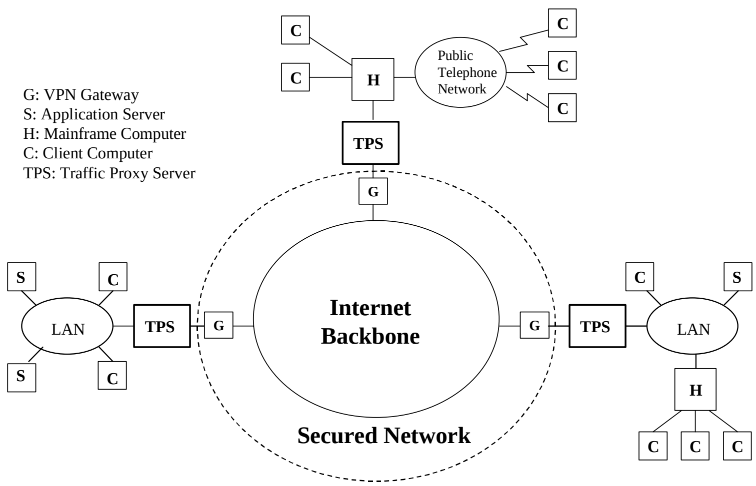


Figure 1: Infrastructure of proxy server-based VPN traffic-pricing system

The protocol for the VPN traffic pricing is a three-stage process:

1) A TPS periodically decides the usage price, based on its current traffic load status, for the bandwidth service between the local network and the Internet. The price and the traffic load status are disseminated to those client computers that are using the VPN transmission services.

2) A smart agent, i.e., a client side application, helps users to estimate jobs' values and sizes, as well as the effects of delay on the jobs. It automatically makes job submission

[4] *Job* is considered a legacy term from the old batch-processing system. We adopt this name to keep the consistency with other papers, having used it previously.

decisions according to the net values of the jobs by taking account of transmission service prices and throughput times. These decisions are imprecise, but statistically the errors are acceptable because users can update the rules for job size predictions and job value estimations, and therefore improve the smart agent's performance.

3) Once a job is submitted it will be charged with the price by the real size of total data flows serviced by the VPN. The costs are billed to the user's account.

We suggest a two-session interactive model for TPS operation. A similar model has been designed in practice for secure proxy servers (SPSs) [23]. One of the sessions is designed for security control purposes and another for data transmission service. The following process provides an example for a data retrieval transaction:

- A remote client on a user’s behalf requests a connection to the TPS with the user’s ID and password.
- If the user’s information is properly authenticated, the client is permitted to proceed to request information services from the application server that the TPS proxies. Meanwhile the TPS checks an active user list. If the user/client pair is not on the list, the TPS adds the user/client to the list. The TPS periodically sends pricing information to all active clients according to the active client list, so that users are able to make job submission decisions.
- The user may submit jobs after the primary control connection is properly processed. Only the jobs with positive net values are submitted, which are calculated by a smart agent using the expected throughput time and the price that the client received from the TPS through the primary connection. In this stage, the client establishes a secondary connection to the TPS.

- The TPS authenticates and authorizes each job request and relays it to a destination application server.
- The application server completes services and sends back flows of data to the TPS.
- The TPS schedules data transmission tasks for the affiliated VPN gateway and bills the services to the user's account according to the jobs' sizes and the user's QoS requisitions.
- The TPS forwards data flows to the client.

The major difference between the interaction model suggested for the TPS and the one used for the SPS is that the TPS schedules data flows and accesses user accounts each time that an application server generates data flows, while the SPS function is accessed before a service starts. Some protocols, such as *Remote Authentication Dial-In User Service* (RADIUS) [25][26] that has been used by some VPN solutions, can be utilized for the TPS in user authentication and remote network connection administration.

3. A Dynamic Traffic-Pricing Model for the VPN with M/G/1 Queueing

Several restrictions are necessary to narrow down the VPN traffic-pricing problem. First, we focus our discussion on the data flows incurred by application service requests but neglect the traffic for the routing and addressing services, such as domain name service and routing information distribution, because they have relatively less impact on the network bandwidth. Second, in the main part of this paper we will discuss the traffic-pricing model based on the assumption that a job incurs a single data flow. We can extend the result to a general situation, where one job is related to multiple data flows[5]. This strategy allows us to simplify the model derivation without losing the correctness of the outcome. Third, we do not consider traffic management in the backbone because it is out of the organization's control. We assume that ISPs assure the requested QoS on the Internet in accordance with the contracts, such as a service-level agreement. Fourth, although the congestion caused by application servers may likely happen, we assume it can be solved as a separate issue so we can focus our discussion on the transmission problem. Finally, traffic control within LANs is not considered. This is because LANs normally have enough bandwidth and organizations are more interested in the network bottlenecks located at LANs' Internet connections, which are blamed for the traffic congestion that can be observed at VPN gateways.

3.1 Characterizing the VPN Bandwidth Service

A network's transmission capacity depends on the bandwidth of circuits between nodes. We assume all these circuits are full-duplex. That is, every circuit provides a pair of transmission services that send data flows in opposite directions. We call a one-way transmission service for a data flow as *a channel*. *A route* is made up of a series of network nodes and channels and is a one-way circuit for a data flow.

We can define an Internet-based VPN consisting of a set of $G = \{g_s\}$ gateways, $s \in S$, and an equal number of channels $C = \{c_s\}$ connecting gateways to the Internet, providing that each gateway has only one Internet connection. An Internet *tunnel* for a pair of specified VPN gateways is a route between the correspondent Internet routers at another end of the Internet channels for the gateways . The service quality of the tunnel as well as the channels depends on the service-level agreement between the organization and the Internet service provider.

---

[5] The extension is straightforward under the assumption of sequential processing of the multiple data flows, since the costs are additive. The proof is somewhat tedious and trivial, so it is not added as another appendix but is available upon request.

With the assumption of the assured Internet QoS the bottleneck of a VPN route, if it exists, is one of the two channels between VPN gateways and the Internet, not the Internet tunnel. Consider a route R^j carrying data flow *j* from a sender to a receiver. R^j is a set of nodes and channels. It can be denoted as $R^j = \{g_1^j, g_2^j, c_1^j, c_2^j, c_n\}$, where g_1^j is the gateway with a bottleneck Internet connection c_1^j, g_2^j is the gateway with a non-bottleneck Internet connection c_2^j, c_n^j is an Internet tunnel for data flow *j*. g_1^j and $g_2^j \in G$, c_1^j and $c_2^j \in C$. Here, we say channel c_s^j is a bottleneck in route R^j referring to a data flow *j* if the channel's cumulative available bandwidth capacity is always less than another channel's cumulative available bandwidth capacity in the time period servicing the data flow. If the data transmission request at a VPN gateway is a Poisson process with a general size distribution, the bandwidth allocation service becomes an M/G/1 queueing system because the capacity of a channel is deterministic but the size of the data flows varies. [14]

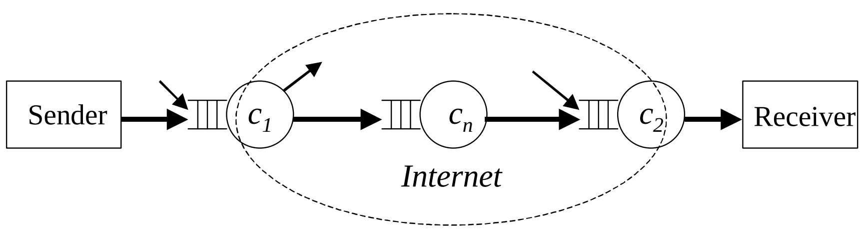


Figure 2: A three-stage queueing model for a VPN route.

Generally, the above VPN route can be modeled as a three-stage queueing system (Figure 2): the first queueing server is the Internet connection channel c_1 for outgoing data flows from the sender; the second one is the Internet tunnel c_n; and the third one is the Internet connection channel c_2 for incoming data flows to the receiver. The traffic loads on channel c_1 also come from other computers in the LAN the sender locates. A major portion of outgoing data flows from channel c_1 is sent to other destinations than the receiver being discussed, and therefore they will not join the queue for channel c_2. In a similar way channel c_2 also transmits data flows from other sources to the LAN the receiver locates without increasing the burden on channel c_1.

Even though the capacity of the Internet connection channels is the main concern in traffic control, the channels are passive to traffic loads, and routers/gateways are actually allocating bandwidth for them. The traffic through channel c_1 can be observed and measured at gateway G_1 that maintains the queue for channel c_1 and allocates the bandwidth for c_1. Although the queue for channel c_2 forms at a router in the Internet, the traffic through channel c_2 can still be observed and controlled at gateway G_2. In this context, the VPN gateway and the transmission channel are equivalent in traffic control. If an Internet connection channel is the bottleneck in a route, we can also refer to the affiliated VPN gateway as the bottleneck gateway.

In the transaction-level pricing architecture, the TPS replaces the VPN gateway to allocate the bandwidth of Internet connections. We investigate two bandwidth-scheduling algorithms, prioritized first-in-first-out (FIFO) bandwidth scheduling and non-prioritized round-robin (RR) bandwidth scheduling. In prioritized FIFO scheduling, the priority is one of the user's choices in data transmission services, which makes a difference to the delay cost: if the queue of a higher priority is nonempty the service for the queue of a lower priority will never begin. In another aspect, we assume there is no preemption once a data flow transmission has begun. That is, if a data flow at a lower priority is being serviced and another data flow at a higher priority arrives, the data flow at the lower priority is continued until completion. A higher priority service normally costs more than a lower priority service, but it takes less time to complete, meaning a lower delay cost. Therefore, there is a trade-off between a better service quality and a lower service cost. Within the same priority class, the bandwidth service complies with a regular FIFO scheduling.

With non-prioritized RR bandwidth scheduling, a VPN allows all data flows being serviced to share Internet connection channels equally. When a data flow requires the transmission service of a VPN channel, it joins the tail of the queue and is assigned a fixed-length time slice for using the transmission service in its turn. This transmission service policy implies that all data flows are serviced whenever they arrive at the node. Obviously, the expected elapsed time between two consecutive service time slices for this data flow is the number of data flows in the queue being serviced multiplied by a slice of time.

The packet-switched technology used by the Internet provides cost-effective services for VPNs. One of the important advantages of packet-switched networks is that the total delay a data flow encounters in a route is less than the summation of waiting and servicing times in all channels (or equivalently nodes) along the route. This feature allows packet-switched networks to provide better bandwidth efficiency than other types of network, such as circuit-switched networks. However, the "pipeline" effect in a packet-switched network adds complexity in setting up the price for traffic because the total throughput time is not a sum of the throughput times at each individual packet-forwarding device. The same effect is applied to VPNs built on the Internet.

Proposition:

In a packet-switched network, the total throughput time that a data flow is transmitted through a route can be expressed as the time spent at the bottleneck node plus a relatively trivial amount of delay on other nodes of the path.

The following is an intuitive explanation of the proposition:

When a data flow is transmitted through a route in a packet-switched network, packets are sent just like water flows through a pipe. Every node in the route forwards the packets through the channel to next node whenever it receives the packets and the channel has available bandwidth. Because of time overlapping, the total throughput time for the data flow is less than the summation of throughput times at each node. By intuition, the time that the data flow passes through the bottleneck node will almost overlay the throughput time at any other node. There is only a minor difference in transporting a unit of the data flow, which has been defined as a segment of packets. The number of packets in a segment depends on the flow control algorithm adopted by the network. Therefore, the total time of the data flow transmitted through the route can be expressed as the throughput time for the bottle node plus the time for a unit of packets to be sent through the route.

Applying the above proposition to the VPN, we can infer that the total transmission time for a data flow through a VPN route is the throughput time of the bottleneck Internet connection plus the time for a segment of packets, which is the processing unit of packets, to go through the other path. Every segment is assumed to have the same size.

3.2 Deriving The Optimum Pricing Formula

The following economic conditions are assumed for the VPN traffic-pricing model:

- A job has an intrinsic value perceived by the user who generates it;
- A job's net value to the user depends on three factors: the intrinsic value, the price charged, and the delay cost which is proportional to the throughput time;
- Prices are set for the channel bandwidth consumption in accordance with the network traffic status;

- The information of both the bandwidth price and the expected throughput time are periodically disseminated to users;
- Users are rational, i.e., they submit their jobs only if expected net gains from job submissions are positive.

It must be emphasized that each user makes job submission decisions independently. A longer throughput time has a higher negative effect on the job's value, and vice versa. The problem on the client's side is that there is limited and imprecise information available to make a decision. The available information includes current price, the predicted throughput time issued by the network, and realized utilities of previously executed jobs. Although the network sets service prices for each section of circuits according to the expected throughput time of each channel, distributions of the job size and the average delay cost; a user only sees the price as the consequence of the job size and the service quality chosen for the job. The user is aware that after a job is submitted, its real throughput time is a random outcome depending on the momentary traffic condition when the job is being serviced. Individually, the user's job has little effect on the price. Therefore, every user is more concerned with a job's net value and does not worry about the consequences of his/her choice on the aggregate outcome.

Exogenous Demands for Transmission Services

From the user's viewpoint, we define *job type* to distinguish the characteristics of a job, such as job size and application type. Different users may submit the same types of jobs at different moments. However, jobs of the same type remain identical in size and application type, even though their values vary from user to user and from time to time. We model the demand for transmission services as determined by (i) an exogenous "potential" rate which would prevail if all costs were zero, and (ii) the monetary and time costs of obtaining the service. Let *I* denote the set of users, *J* the set of job types, and *Q* the set of job sizes. Let λ_{ij} be the exogenous potential service request rate of job type *j* from the user *i*. The total exogenous rate, i.e., the maximum potential job rate, for the VPN is $\lambda = \sum_{i=1}^{I}\sum_{j=1}^{J}\lambda_{ij}$, and the maximum potential data flow rate is $\sum_{j=1}^{J} q_j \sum_{i=1}^{I}\lambda_{ij}$, where $q_j \in Q$ is the size of a type *j* job measured in the number of segments. These two forms of exogenous rates are based on the aggregation of the demands when there are no costs—no transmission charges and no response delays that impose extra costs to services. A cost reduces a service's net value and hence prevents the submission of lower-valued jobs. Therefore, in reality, not every potential service request will be submitted because the costs from service delays are inevitable. The real demands in responding to the price and the delay time are the rates of jobs submitted to the network.

Transmission Delay Costs and the User Utility Function

The utility of a job is defined as the net value of the job determined by the job's gross value, the delay cost, and the bandwidth price in a network system with traffic pricing. A type-*j* job's gross value V_{ij} reflects the benefit that user *i* receives in the case of no cost. V_{ij} is a random variable depending on user *i*'s evaluation at the moment when the type-*j* job is submitted. It includes the effect of application service time at the application server that is assumed not a bottleneck in this model. The network transmission service delay results in the disutility of jobs and its coefficient is denoted as δ_{ij}. δ_{ij} is also a random variable relevant to the moment user *i* submits a type-*j* job. V_{ij} and δ_{ij} have a joint density function $f(V_{ij}, \delta_{ij})$.

Denote t_{jks} the expected time for type-j job to get through channel c_s with priority k service, and d_j the time for a single segment of type-j job to get through all other channels. If channel c_s is not in the route of data flow j, then $t_{jks} = 0$. In a VPN d_j represents the unit time for a type-j job to go through the Internet tunnel in the route. The total expected throughput time of type-j job with priority k, τ_{jk}, is a function of $\{t_{jks}\}$ and d_j.[6] We assume the expected throughput time is linear in regard to the size of the job without considering packet tossing when network traffic load is under control. According to the proposition in Section 3.1, τ_{jk} can be expressed as:

$$\tau_{jk} = t_{jkH} + \frac{1}{q_j} t_{jkL} + d_j \qquad (3.1)$$

where t_{jkH} is the throughput time at the bottleneck channel c_H^j and $\frac{1}{q_j} t_{jkL}$ is the average unit throughput time at the non-bottleneck channel c_L^j, with c_H^j and $c_L^j \in R^j$.

Since the delay caused by the bottleneck channel c_H^j dominates the total throughput time of the transmission service of the route, the price of the service provided by the bottleneck channel is the main factor affecting the user's behavior because it directly controls the data traffic through the channel.

<u>Users' Choices</u>

In a network without traffic pricing, only the delay cost reduces a job's net value. Thus, the net value of a type-j job with a gross value V_{ij} submitted by user i with a priority k service class request is:

$$V_{ij} - \delta_{ij}\tau_{jk}, \ \forall i, j, k \qquad (3.2)$$

[6] As defined in last paragraph, this throughput time does not include the service time at the application server.

In a priced network the price is another cost to a job's service. We adopt a pricing policy in which a price is set according to the job's size, service quality request, and traffic conditions. Also the price does not depend on job type and the owner of the job. Denote the price set as $r = \{r_{qks}\} = \{r(q, k, s); q \in Q, k \in K, c_s \in C\}$, where $r(q, k, s)$ is the price for a job of size q serviced by channel c_s with priority class k. The size of the type-j job is normally denoted as q_j, because more than one type of job may have the same size. Then the net value of a type-j job submitted by user i requesting for a priority k service becomes $V_{ij} - \delta_{ij}\tau_{jk} - r_{jk}$, where $r_{jk} = r_{qkH} + r_{qkL}$ (with $q = q_j$) is the total price paid for job j transmission by two Internet channels of the VPN in the route.

Assume that before making a submission decision for job j, user i is able to obtain both the price and the transmission delay time information for the VPN. If there exists a k such that $V_{ij} - \delta_{ij}\tau_{jk} - r_{jk} \geq 0$, then the user submits the job. The user will maximize the job's utility by choosing a priority

$$k^* = sup_{k \in K}\{ V_{ij} - \delta_{ij}\tau_{jk} - r_{jk} \} \text{ s.t. } V_{ij} - \delta_{ij}\tau_{jk} - r_{jk} \geq 0.$$

Define a user i's job submission distribution π_{ijk} as:

$$\pi_{ijk}(V_{ij}, \delta_{ij}; r, \tau_{jk}) = \begin{cases} 0 & u_{ijk^*} < 0, \ or\ k \neq k^* \\ [0,1] & u_{ijk^*} = 0, \ and\ k = k^* \\ 1 & u_{ijk^*} > 0, \ and\ k = k^* \end{cases} \qquad (3.3)$$

Given job j's value V_{ij}, its delay cost coefficient δ_{ij}, bandwidth price set r and the expected throughput time set $\{t_{jks}\}$, the expected net value of the job to be redeemed by user i based on job submission choice π_{ijk} is:

$$(V_{ij} - \delta_{ij}\tau_{jk} - r_{jk})\,\pi_{ijk} \qquad (3.4)$$

The Optimum Price

With (3.3), the exogenous traffic rate λ_{ij} is translated into an average arrival rate x_{ijk} in the VPN for priority class k service as the following:

$$x_{ijk} = \lambda_{ij} \iint_{V_{ij},\delta_{ij}} \pi_{ijk}(V_{ij},\delta_{ij};r,\tau_{jk}) f(V_{ij},\delta_{ij}) dV_{ij} d\delta_{ij} \qquad (3.5)$$

Denote $\pi = \{\pi_{ijk}\}$ as the user's submission choices and $X(\pi) = \{x_{ijk}(\pi_{ijk})\}$ the set of the traffic rate for all jobs submitted to the VPN, $i \in I$, $j \in J$ and $k \in K$. The average arrival rate of size-q jobs for the priority k service of channel c_s is expressed as $\varphi_{qks} = \sum_{j \in J} \mu_{jqks} \sum_{i \in I} x_{ijk}$, where $\mu_{jqks} = 1$ if type-j jobs go through channel c_s with size q and priority k service, otherwise $\mu_{jqks} = 0$. The job arrival rate with priority k in channel c_s is $\sum_{q \in Q} \varphi_{qks} = \varphi_{ks}$. The total job arrival rate in channel c_s is $\sum_{q \in Q} \sum_{k \in K} \varphi_{qks} = \varphi_s$. Denote $\varphi(X) = \{\varphi_{qks};\, q \in Q,\, k \in K,\, c_s \in C\}$.

The expected throughput time for a type-j job processed by channel c_s with priority k can be expressed as a function of the distribution of arrivals by size and the bandwidth: $t_{jks} = \Omega_{jks}(\varphi(X), B_s)$, where B_s is the available bandwidth of the channel c_s. Then, the overall benefits of the network service can be expressed as a welfare rate maximization problem in conjunction with (3.3):

$$W(\pi,\tau) = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \lambda_{ij} \iint_{V_{ij},\delta_{ij}} [V_{ij} - \delta_{ij}\tau_{jk}] \pi_{ijk} f(V_{ij},\delta_{ij}) dV_{ij} d\delta_{ij}$$

$$\text{s.t. } t_{jks} = \Omega_{jks}(\varphi(X), B_s) \qquad (3.6)$$

Applying the Lagrangian approach we can obtain the following conditions for maximizing problem (3.6). These conditions are necessary and sufficient for VPN route R^j servicing type-j jobs submitted by user i:

$V_{ij} - \delta_{ij}\ \tau(q_j, t_{jkH}, t_{jkL}, d_j)$

$$\geq \sum_{l\in I}\sum_{m\in J}\sum_{h\in K} [\frac{\partial\Omega_{mhH}}{\partial\varphi_{q_j kH}} + \frac{1}{q_m}\frac{\partial\Omega_{mhL}}{\partial\varphi_{q_j kL}}]\lambda_{lm} \iint_{V_{lm},\delta_{lm}} \delta_{lm}\pi_{lmh} f(V_{lm},\delta_{lm}) dV_{lm} d\delta_{lm}$$

and $\lambda_{ij} f(V_{ij},\delta_{ij}) > 0$ *(3.7a)*

or VPN route R^j not servicing any type-j job:

$V_{ij} - \delta_{ij}\ \tau(q_j, t_{jkH}, t_{jkL}, d_j)$

$$< \sum_{l\in I}\sum_{m\in J}\sum_{h\in K} [\frac{\partial\Omega_{mhH}}{\partial\varphi_{q_j kH}} + \frac{1}{q_m}\frac{\partial\Omega_{mhL}}{\partial\varphi_{q_j kL}}]\lambda_{lm} \iint_{V_{lm},\delta_{lm}} \delta_{lm}\pi_{lmh} f(V_{lm},\delta_{lm}) dV_{lm} d\delta_{lm}$$

and $\lambda_{ij} f(V_{ij},\delta_{ij}) = 0,$ *(3.7b)*

where Ω_{mhH} is the throughput time at bottleneck channel c_H^m in route R^m and Ω_{mhL} is the throughput time at non-bottleneck channel c_L^m in route R^m.

Let $x_{lmh}\bar{\delta}_{lm} = \lambda_{lm} \iint_{V_{lm},\delta_{lm}} \delta_{lm}\pi_{lmh} f(V_{lm},\delta_{lm}) dV_{lm} d\delta_{lm}$, where x_{lmh} is the real submission rate of type-m jobs by user l and $\bar{\delta}_{lm}$ is the delay cost coefficient weighted by submission rate. According to Gupta, Stahl and Whinston [10], (3.7) implies that the benefit-maximization rental price expressed as the following:

$r_{qks}^* = \sum_{l\in I} \sum_{m\in J}\sum_{h\in K} \frac{\partial\Omega_{mhs}}{\partial\varphi_{qks}} x_{lmh}\bar{\delta}_{lm}$, if channel c_s is the bottleneck of route R^j; *(3.8a)*

$r_{qks}' = \sum_{l\in I} \sum_{m\in J}\sum_{h\in K} \frac{1}{q_m}\frac{\partial\Omega_{mhs}}{\partial\varphi_{qks}} x_{lmh}\bar{\delta}_{lm}$, if channel c_s is **not** the bottleneck of route R^j. *(3.8b)*

We change the subscript of φ for job size from q_j to q because the price is relevant to job sizes, not to job types. Generally, $q_m >> 1\ \forall m \in K$, hence $r_{qks}^* >> r_{qks}'$. Then we can approximately apply a single price r_{qks}^* to type-j job submitted by user i for service priority k neglecting the effect of the non-bottleneck price r_{qks}'.

The Traffic Pricing Formula for FIFO Bandwidth Scheduling

Since the bottleneck price of a VPN Internet channel is only affected by the local traffic, subscript *s*, which indexes the channel, is negligible by default. If a data flow is not indexed by a channel label *s*, it should be interpreted as the traffic through the pricing channel being discussed. Therefore, in the follow-up discussion we discard the subscript *s* from all notations.

The total price paid for a job *j* at priority *k* of the channel can be expressed as:

$$r_{jk} \approx r^{*}_{qk} = \sum_{l \in I} \sum_{m \in J} \sum_{h \in K} \frac{\partial \Omega_{mh}}{\partial \varphi_{qk}} x_{lmh} \bar{\delta}_{lm} \,. \qquad (3.9)$$

In the above pricing formula, the $x_{lmh}\bar{\delta}_{lm}$ term is the flow cost of delay for each user of the Internet channel via the VPN gateway, and the $\frac{\partial \Omega_{mh}}{\partial \varphi_{qk}}$ term is the marginal time delay caused by an extra type *j* job at priority *k*. Thus, the optimal resource allocation is achieved when a type *j* job at priority *k* pays the service for the marginal increase of the aggregate flow cost of delay for all users. The price will be high when the traffic is heavy and/or the cost of time is high.

With FIFO scheduling, the expected throughput time t_{jk} depends on the expected waiting time in the queue, w_k, which is invariant to a job's type, and the channel service time, which is invariant to priority class once the job is being serviced. Therefore, the queue waiting time w_k for a M/G/1 queueing system can be expressed in terms of job size *q* instead of job type *j* [10]:

$$w_k = \frac{\sum_{h \in K} \sum_{q \in Q} \varphi_{qh} q^2}{2B^2 (1 - \sum_{h<k} \rho_h)(1 - \sum_{h \le k} \rho_h)}$$

Then the throughput time:

$$\Omega_{jk}(\varphi(X), B) = w_k + q / B \qquad (3.10)$$

By using the above formula, we can obtain a pricing formula for a FIFO scheduling that is quadratic in job size:

$$r^*_{qk} \;=\; \sum_{h\in K}\varphi_h\bar{\delta}^{(h)}(a_{1h}q + a_{2h}q^2) \quad \forall q \in Q, k \in K \qquad (3.11)$$

where $a_{1h} = \dfrac{w_h(2-\sum_{l<h}\rho_l - \sum_{l\le h}\rho_l)}{B\,(1-\sum_{l<h}\rho_l)(1-\sum_{l\le h}\rho_l)}$ when $k < h$, $a_{1h} = \dfrac{w_h}{B\,(1-\sum_{l\le h}\rho_l)}$ when $k = h$,

$$a_{2h} = \frac{1}{2B^2(1-\sum_{l<h}\rho_l)(1-\sum_{l\le h}\rho_l)},$$

ρ_l is the bandwidth utilization ratio for priority l service and $\bar{\delta}^{(h)} = \sum_{l\in I}\sum_{m\in J}\dfrac{x_{lmh}}{\varphi_h}\delta_{lm}$ is the mean of δ_{lm} over user l and job type m, weighted by the flow in priority h.

The Traffic Pricing Formula for RR Bandwidth Scheduling

We next consider the pricing formula for a non-prioritized RR scheduling scheme. The subscript k can be eliminated for the non-prioritized case. In a RR scheduling system, a job's throughput time—the time that a job stays in a VPN gateway—is proportional to job size and the average number of jobs in the queue during servicing. The throughput time of a size q_j job in a non-prioritized queue is $t_j = t(q_j) = [(L^*+1)\, q_j - \rho/2]/B$, where $L^* = \dfrac{\rho^2 E[q^2]}{(1-\rho)(E[q]+E[q^2])}$ is the average number of jobs being serviced, ρ is the bandwidth utilization ratio of the gateway, and $E[q]$ and $E[q^2]$ are the expected size and size-squared values for the gateway (see Appendix).

The type-j job submission criterion for user i is $q_j(v_{ij} - \delta_{ij}t_j) \ge r_j$, or $v_{ij} - \delta_{ij}t_0 \ge r_0$, where $v_{ij} = V_{ij} / q_j$ is job's unit value, $t_0 = t_j / q_j$ is the unit throughput time, and $r_0 = r_j / q_j$ is the unit price. We can derive the optimal unit price:

$$r_0 \approx \frac{\bar{\delta}}{B}(L^* + \frac{L^* - \rho}{1-\rho}), \qquad (3.12)$$

where $\bar{\delta} = \sum_{l\in I} \sum_{m\in J} x_{lm}\delta_{lm}q_m / \sum_{l\in I} \sum_{m\in J} x_{lm}q_m$ is the mean of δ_{ij} over i and j, weighted by data volume rates. The approximate form for the pricing with RR scheduling indicates that the expected number of jobs in the queue is a critical factor in a job's price and the price is proportional to the size of a job.

4. Experiments of VPN Traffic-Pricing

4.1 Experiment Design

VPN Traffic-Pricing Experiment System

We developed a prototype system named *VPN Traffic-Pricing Experiment System* (VTPES) [18] to test the transaction-level pricing architecture and to conduct experiments for the pricing model. VTPES is built on a small network platform in the Center for Research in Electronic Commerce (CREC) at UT Austin. It has three major distinctions from the previous GSW model simulation system:

1) It is a real-time system. Both data traffic generation and bandwidth allocation are implemented on a real-time basis.
2) It is a scalable distributed system. VTPES runs with a real network consisting of several computers. The configuration of the network can be varied to test the performance of VPN traffic pricing under certain definable conditions.
3) It has a dual-queue structure. In addition to a regular queue structure for bandwidth services, an extra queue is configured as a benchmark system. By using a shared traffic

generation source, we can test different experimental schemes and compare the outcomes with that of a standardized scheme.

A complete system of VTPES consists of five components representing a network route: a client computer, a client-side VPN gateway, a server-side VPN gateway, a TPS, and an application server (Figure 3). The client computer generates jobs and submits those jobs that are expected to create net benefits after taking out all costs. The client-side VPN gateway routes the jobs to the destination application server via the server-side VPN gateway. The application server services the jobs relayed by the TPS and generates data flows back to the client computer. The TPS performs pricing, bandwidth scheduling and user-access accounting, etc.

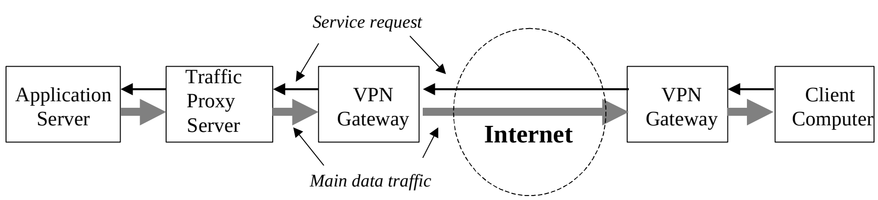


Figure 3: A unit VPN traffic-pricing system for experimentation

This 5-tier pricing system can be reduced to a smaller scale to ease the experiment without losing the desired experimental features. First, the client-side gateway is “transparent” to data flows and can be ignored. Second, the server-side VPN gateway’s bandwidth scheduling function can be merged to the TPS because the TPS and the gateway can be logically considered as an integrated subsystem. Thus, the server-side gateway can also be ignored, handing over its functions to the TPS. In fact, the terms, VPN gateway and TPS, are exchangeable when referring to the experiment components. Third, the role of an application server is also trivial in data transmission simulations, because the application server is assumed not a bottleneck and the application service time is irrelevant to the network transmission time of a data flow, meaning it has no effect on bandwidth pricing.

Finally, a simplified network was configured containing a minimum of two physical computers connected via 10 Mbps Ethernet. In this two-tier network configuration (Figure 4), one computer runs a module called the *Traffic Load Generator* (TLG) to produce service requests. Another computer acts as the TPS, the server-side gateway, and the application server.

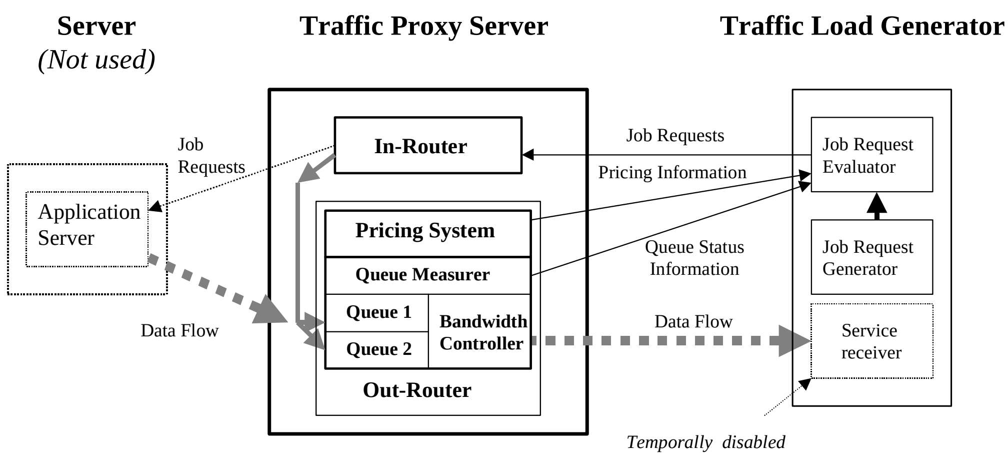


Figure 4: Modules building up a unit traffic-pricing system

The 10 Mbps Ethernet connection between two computers is controlled by a software model to simulate an Internet tunnel for the experiment system. The tunnel bandwidth is configurable from 14.4 Kbps to 10 Mbps to meet experiment requirements. The Bandwidth Controller sends data flows in small blocks to fit a time slice of 0.02 seconds adapting to the preset bandwidth parameters. The actual data transmission rate is measured every second and fed back to the Bandwidth Controller. Then the Bandwidth Controller adjusts the size of data blocks being transmitted with regard to the detected errors. In this way, the accuracy of bandwidth can be controlled with an error less than 0.03%.

One more consideration is how to ease client-side data receiving that produces the final service benefits for submitted jobs. As an alternative option, VTPES allows the TPS computer to run a data-receiving module that originally belongs to the client computer. Thus, all measurements can be centrally done and controlled at the TPS. This simplification option is for experiment data collection only and will not lower the accuracy of data measurements as comparative experiments have proved.

Traffic load generation

Two concepts of data traffic have been mentioned: exogenous data traffic and the arrival data traffic. The former is the maximum potential demand of the network generated by TLG, which is actually unobservable in the real world. The latter is the true rate through the VPN gateway, which is the aggregate of the submitted jobs. We denote these two types of traffic as *job generation rate* and *job arrival rate*.

TLG can generate data traffic according to the job size distribution and the job request rate defined in a profile. The arrival of jobs can comply with Poisson process.[7] TLG can operate in either *dynamic mode* or *pattern-based mode*. In the dynamic mode, the intervals of consecutive jobs are dynamically generated. A feedback mechanism is optional to maintaining the deviation of the exogenous job-generation rate within 1.5-3% to guarantee the accuracy of experiment. This rate will be translated to a real traffic rate to be processed by the TPS. In the pattern-based mode, the series of job generation intervals is pre-generated in a desired distribution and saved as a parameter file to be used by TLG. This mode has better comparability for time series data collected from different experiment schemes. The job generation rate of this mode is not controllable due to the rigidity of the preset data series pattern.

[7] This implies the job interarrival times are exponentially distributed, and independent. For simplicity, we verified the stochastic properties by 1) examining the arrival-time independency of consecutive job requests, and 2) observing the histogram of the interarrival times. The regression on the data series demonstrates the independence between consecutive time intervals. The ordinary least squares (OLS) regression on 10-period lags of job request arrival intervals with 2560 samples has an adjusted R^2 of 0.025, and the OLS regression on 2-period lags of job request arrival intervals with 2568 samples comes with an adjusted R^2 of 0.009. The low values of adjusted R^2 suggest that the correlation between two consecutive arrival intervals is insignificant. Also the logarithm of the histogram from the job interarrival times is linear as well. Detailed explanation in techniques for testing Poisson process can be found in Appendix A of [25] by Paxson and Floyd.

Since the job arrival rate at the TPS together with job sizes jointly imposes traffic loads at the TPS, the job size distribution is a critical parameter in TLG configuration. TLG also provides both dynamic and pattern-based modes for generating job size distributions.

System timing control

Timing is important for VTPES to control its experiment processes. VTPES uses four parameters for timing control: the system update interval, the data sampling interval, the window size for averaging observations, and the pricing interval. The system update interval is the size of time slice in which VTPES services jobs and updates the queue length. A time slice that is too large will reduce the accuracy of data measurements and a time slice that is too short means more computation overhead for the TPS. It is currently set at 0.02 of a second per time slice. The data sampling interval is the time elapse between two consecutive system status measurements. Similar to the system update interval configuration, a higher frequency of measurement will result in higher accuracy, but costs more CPU overhead. At present, we choose 5 times per second queue status sampling, covering the queue length and the number of queueing jobs. Properly choosing the window size for averaging observations can provide VTPES a stable and yet effective data series for pricing decisions; and the price must be applied to a sufficient duration in which VTPES can smooth the instability caused by over-reactions to a fluctuating traffic condition. The relationship between the four timing parameters is: system update interval << sampling interval << moving averaging time window size << pricing interval.

System status measurement and data collection

The data needed for analyses include the observations measured during the experiment process and the one derived from the originals. VTPES can dynamically collect a number of time series data including those collected per second, those collected when a new job arrives, and those collected after a job has been serviced. Typical examples are the instantaneous values of queue length, the number of jobs in the queue, price, a job's expected service benefit and real benefit after service, etc. The derivations are mostly the average of the raw data on per second and per pricing-interval bases. The size of the observation window ranges from 1400 seconds to 5000 seconds depending on need. We carefully discarded those observations obtained before the system reached a steady state. Normally, the observation window starts after 1000 seconds when the queue length in the TPS computer reaches an equilibrium state. Also we dropped the last few observations to avoid the possible inaccuracy of the measurement when the operation of the experiment system was about to terminate. All these data are saved instantly.

4.2 Pricing Effectiveness

Pricing effectiveness is the performance change of the VPN when the pricing mechanism is enacted using the pricing formula obtained in section 4. We mainly examined the welfare rate, i.e., per-second service welfare that is the output being monitored, to evaluate the effect of different scheduling schemes for non-priced and priced systems on VPN bandwidth service performance. The network load parameters for the experiment, i.e., the input, include the distributions of job size, the job value, and the delay cost coefficient. The job value and delay cost are random variables depending on the user’s timely preferences and can be generated by the computer. Job size can be predefined in a profile.

In order to test whether the performance from the experiment based on a real network is the same as that from the GSW software-based simulation, we intensively used the same parameter values for most experiment schemes as those used in the GSW simulation. Although we also designed experiments with different job parameter distributions, using GSW experiment parameter values allows us to compare the data obtained from our experimental schemes with the ones from GSW model experiments.

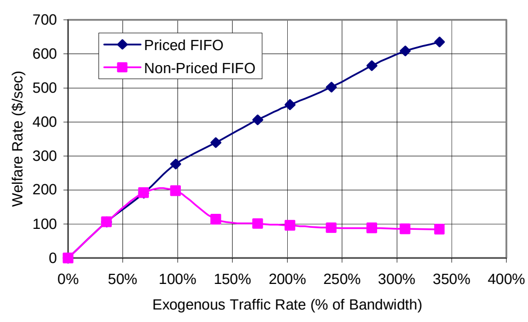
The job value distribution used in the GSW model simulation has a normal distribution with a mean of 500 dollars and a standard deviation of 150 dollars, i.e., job value ~ $N [500, 150^2]$. It is independent of job size. The delay cost coefficient distribution is also independent of job size. The GSW model uses a normal distribution for the delay cost coefficient with the mean of 4 dollars and the standard deviation of 1 dollar, i.e., job delay cost ~ $N [4, 1]$. The absolute levels of the job value and the delay cost coefficient are not important, but the relative levels and their correlation to job size are critical. Therefore, in addition to this basic set of distributions, we also tested two diversities: a set of job parameters with different standard deviations, and a set of the parameters that are relevant to the job size.

We conducted this group of experiments in four steps:

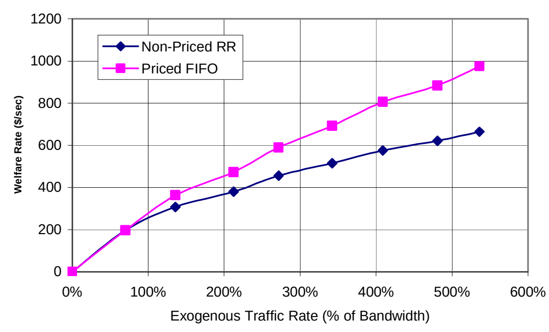
1) Verify the consistency of the outcome from a network-based experiment and the one from the GSW model simulation using FIFO scheduling;
2) Test the performance of the VPN using non-priced RR scheduling;
3) Compare the outcomes of priced schemes between RR scheduling and FIFO scheduling; and
4) Test the priced schemes using different job parameters.

The outcome from the simulation running on VTPES using the FIFO scheduling matches that from the GSW model simulation very well. VPN traffic pricing significantly improves network welfare rate in FIFO scheduling (Figure 5a). The curve of the welfare rate from the non-priced FIFO scheme starts to decline when exogenous traffic rate increases and approaches capacity, while that from the priced FIFO scheme keeps going up as the traffic rate increases. This is exactly the same result as the GSW model simulation revealed.

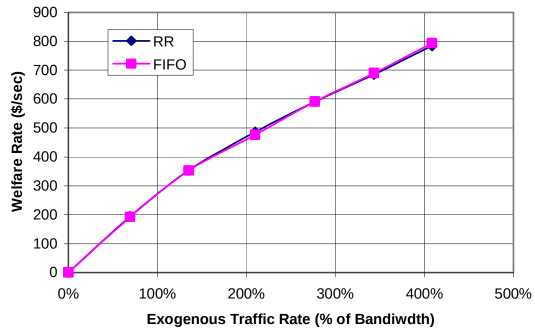
The experiment shows that priced FIFO scheduling performs better than non-priced RR scheduling (Figure 5b). The welfare rate yielded from a non-priced RR scheme increases with the augmented traffic rate. This is because the expected throughput time for a job in RR scheduling is proportional to the job's size. This provides the incentive for users not to submit the jobs with lower unit values. Since the submission decision is based on the comparison between the unit job value and the unit waiting cost, the efficiency of the network is better than that of non-priced FIFO. However, the increase rate of the welfare rate from a non-priced RR scheduling scheme is much lower that that from a priced FIFO scheme. The gap widens when the traffic rate goes up.



(a) Priced FIFO versus non-priced FIFO



(b) Non-priced RR versus priced FIFO



(c) Welfare rate comparison between priced RR scheduling and priced FIFO scheduling

(d) Job submission distribution comparison between priced RR and priced FIFO schemes (exogenous traffic rate: 340% of bandwidth).

Figure 5: Effectiveness of VPN traffic pricing

The experiment demonstrates that pricing is effective in the RR scheduling scheme with approximately the same welfare rate as that from priced FIFO scheduling (Figure 5c). Pricing on FIFO and RR scheduling systems results in almost the same job submission ratio distributions (Figure 5d). In the chart, the *x*-axis is the job size index with larger numbers for larger job sizes and the *y*-axis is the percentage of jobs having been actually submitted in terms of an exogenous job rate. Both the submission ratio distributions consistently drop when the job size is getting larger. The consistency in job submission rate distributions implies that the difference in bandwidth scheduling schemes may not affect users' job submission decisions much when the bandwidth is properly priced.

Although it is necessary to make the outcome of VTPES-based experiments comparable to the previous results from the GSW model, a thorough and complete experimentation should cover more variable factors to provide strong evidence in supporting our conclusions. There are two versions of schemes in this step. The first experiment version uses the parameters with new job value distribution ~ $N\ [500, 200^2]$ and job delay cost coefficient ~ $N\ [4, 2^2]$. As the chart in

Figure 6 shows, the welfare rate obtained from the scheme using the new job value and delay cost distribution has only a negligible difference from that using the parameter values designed for the GSW model simulation.

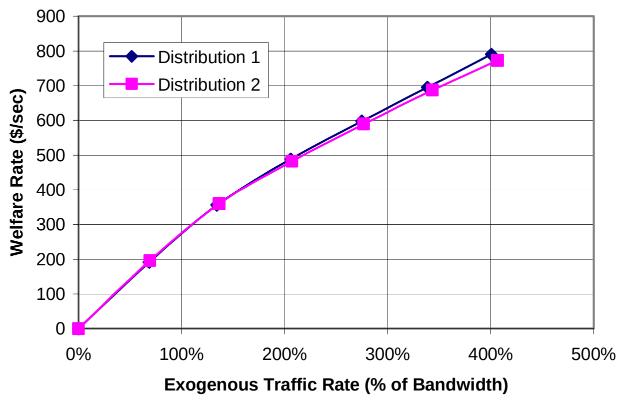


Figure 6: Welfare rates differentiated in standard deviations of job value and delay cost

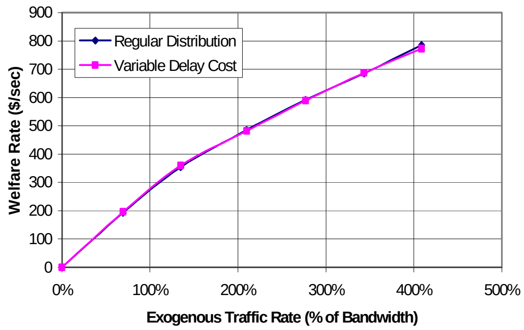
The second experiment version uses a variable delay cost distribution and job value distribution that are relevant to the job size. We use the following conversion formula to make delay cost or job value relevant to job size:

$$X^\wedge = X * (J / B)^{s*0.5}$$

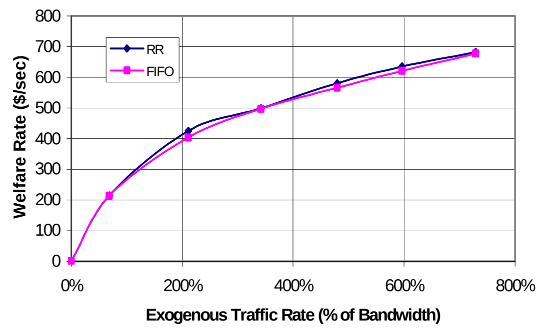
where J is job size, B is bandwidth, $s = 1$ when converting a job value and $s = -1$ when converting a delay cost coefficient, X is a regular value of the delay cost coefficient or a regular job value, and $X^\wedge$ is the converted value. The rationale behind the above conversion formula is that the larger a job's size is, the higher the job's gross value could be or the lower the delay cost coefficient could be. For example, FTP jobs are less sensitive to transmission delays compared to telnet jobs distinguishable in data flow size, or a larger downloading job containing several documents has a higher value than a job simply downloading one of the documents.

The experiment demonstrated that there is no significant difference when using a variable job value or a variable delay cost coefficient correlating to job size. Welfare rate curves from two differentiated schemes in delay cost coefficient, one with a fixed distribution and the other with a variable mean correlating to job size, match well (Figure 7a). The same outcome is obtained from the scheme using variable job value correlating to job size. The curvature of welfare rate curves from the schemes using variable job value is more concave compared to previous welfare rate curves (Figure 7b).

In summary we can conclude that traffic pricing can effectively improve VPN bandwidth service benefits for a wider range of schemes using different job values and delay cost distributions.



(a) Welfare rates from two schemes differentiated in the delay cost correlating to the job size.



(b) Welfare rates from FIFO and RR scheduling schemes using variable job value correlating to the job size.

Figure 7: Welfare rates from the schemes using variable parameters correlating to job size.

5. Conclusions and Future Work

This paper studies the VPN traffic-pricing problem targeting at the practical application with three foci: a transaction-level pricing architecture with a traffic proxy server-based implementation scheme, a dynamic traffic-pricing model for the VPN, and VPN traffic-pricing experiments using a prototype system called VTPES. The transaction-level pricing is proposed for the implementation of the VPN pricing system, taking advantage of VPN's user-account management features. This job-oriented basis is exploited to derive the dynamic VPN traffic-pricing model. The theoretical work is based on the GSW model with two important extensions for the VPN—it is tailored to fit packet-switched networks, typically the Internet, and it uses RR bandwidth scheduling in addition to FIFO bandwidth scheduling. The pipeline effect of packet-switched networks results in the dominance of the bottleneck Internet connection in a VPN route over total throughput time as well as the bandwidth price, which can be controlled by the TPS.

In a simplified case, the total price a job pays for bandwidth services can be approximated by a single price at the bottleneck gateway to reduce the complexity of implementation. In contrast to FIFO scheduling, RR scheduling is closer to real traffic scheduling in use. We revealed that RR scheduling possesses useful implementation features such as allowing a consistent unit price for different types of jobs. Based on these theoretical outcomes we developed a prototype system VTPES to test the implementation scheme and to conduct traffic-pricing experiments. The experimental outcomes from VTPES strongly support the theoretical result. The three aspects of the research in VPN traffic pricing jointly provide a complementary set of solutions for the VPN traffic control problem.

Our ultimate goal in this research is to provide the industry a feasible network pricing architecture that can adapt to current in-use technologies and algorithms for network flow control and queue management. However, there are four apparent limits of our model and implementation scheme. The first one is that a practical gateway may use more than one scheduling algorithm for its bandwidth allocation tasks, for example, a mixed FIFO and RR scheduling. Deriving an analytical form of the price formula for a real gateway is very difficult. Even though the TPS-based solution proposed in this paper may bypass this problem, the mutual effect between the TPS and the VPN gateway in data flow control remains untouched. Therefore, a pricing structure that works without requiring an absolute accurate pricing formula could be a new topic to be researched in next phase.

The second limit is that the proposed VPN pricing model is designed for the problem of the elastic traffic where the optimal objective function counts the total throughput time as the only QoS feature. In pricing real-time traffic, the important factors for evaluating service quality include jitter, i.e., the variation of available bandwidth; committed minimum bandwidth; and maximum allowable bandwidth. Quantifying these features is difficult and more dimensions will hence need to be introduced into the model. One possible solution is to make use of the token bucket algorithm [30] for real-time network traffic pricing. The key is how to feature price curves to optimize QoS parameters in a two-dimension space of bucket size and token rate.

The third limit is inherent in the assumption that the job arrival is a Poisson process. Paxson and Floyd [24] revealed that the packet arrival pattern in the Internet is not a Poisson process. It is apparent that even if the interval of any two consecutive jobs is exponentially distributed the arrival of packets is not necessarily a Poisson process because the number of packets in a job is a random variable and these packets come in batches. Nonetheless, Gupta, Stahl, and Whinston reported in 1999 [11] that the pricing formula based on Poisson arrivals works well even when arrivals are fractal.

The fourth limit is also very critical: we calculate the service delay as a linear function of traffic load at a network node without considering the non-linear effect when congestion happens. Once a network node's buffer overflows, it tosses some packets. This will stimulate reactions of the flow control mechanism to resolve the problem. In this case, the delay of an affected data flow will no longer be linear with its size. Since our experiments have shown that a VPN with the pricing mechanism performs better than a non-pricing VPN when the infinite queue is assumed, the former must also be superior to the non-pricing system with limited queue capacity. In the next research phase, we may let the model cope with the congestion in two aspects. A rigorous approach would be to study a model with finite queue capacity, so the probability of overflow would be explicit, and this probability could be fed back to users who would react in their self-interest. An alternative approach is suggested by the possibility that appropriate pricing may control network traffic so overflows are extremely rare events. These limits reflect the gap between a theoretical research and the practical application. As long as we have a suitable traffic-pricing architecture, the issue raised from the practical system can be solved accordingly with proper patches.

Further research work will be continued in both theoretic and pragmatic aspects. There are two possible refinements for the pricing model. The first refinement is not difficult—the VPN traffic-pricing model can be easily generalized to the one for regular packet-switched networks by assuming that the number of pricing nodes in a route can be more than two. Another refinement of the model is to adopt prioritized RR scheduling for bandwidth allocation, which can work well with the DiffServ architecture. Our non-prioritized RR scheduling lacks the ability to differentiate bandwidth services for different classes of jobs that request different QoS. Although the traffic-pricing model with prioritized FIFO scheduling has the required features, FIFO is seldom used alone for a router or a gateway in practice. A prioritized pricing system can provide choices for users to trade off between better bandwidth service and lower price; and the DiffServ can serve as a framework to support the prioritized scheme. Currently, we are studying Prioritized RR (PRR) [19] and Weighted RR (WRR) scheduling, two special kinds of scheduling schemes. They allow diversified bandwidth allocations for different classes of jobs to maximize their net service benefits.

Empirically, we are going to further exploit VTPES for new experiments with necessary extensions. At present, we use only computer-generated data traffic in the experiments with VTPES, assuming the traffic reflects the real behavior of VPN users. However, due to the availability of the user interface for traffic pricing and human cognitive factors, the outcomes from users may not have the same pattern as we have assumed. We are currently upgrading VTPES for human subject-based experiments. This is to incorporate features such as a user-friendly operational interface, job submission utilities, etc. Our intention is to explore the effects of such a pricing system on the users with budgets and vice versa. This will be in conjunction with other on-going instructional projects in which students utilize a VPN to set up and run electronic businesses. By running this project on our prototype VPN with and without TPS pricing, we will be able to assess not only the network performance under real loads but also the user satisfaction under the various experimental treatments. In another experimentation plan, we will provide students with fundable resources that can be used for accessing our VPN or cashed-in should the student opt to use an alternative non-priced VPN. The objective would be to ascertain if the users perceive a sufficient quality of service on our priced VPN to warrant paying the fee. We will also interview users to ascertain their satisfaction and solicit suggestions for improvements in all aspects of the user interface. This experimental work will guarantee a versatile prototype VPN traffic-pricing system that well meets the requirements of practical industrial applications.

References:

[1] Allman, M., V. Paxson, and W. Stevens. "TCP Congestion Control," Internet RFC2581 (1999), http://www.ietf.cnri.reston.va.us/rfc/rfc2581.txt.

[2] Blake, S. et al. "An Architecture for Differentiated Services," Internet RFC2475 (1998), http://www.ietf.cnri.reston.va.us/rfc/rfc2475.txt.

[3] Braden B. et al. “Recommendations on Queue Management and Congestion Avoidance in the Internet,” Internet RFC2309 (1998), http://www.ietf.cnri.reston.va.us/rfc/rfc2309.txt.

[4] Clark, D. D. “Adding Service Discrimination to the Internet,” *Telecommunication Policy* 20 (1996) 169-181.

[5] Demers, A., S. Keshav, and S. Shenker. “Analysis and Simulation of a Fair Queuing Algorithm,” *Internetworking: Research and Experience* 1 (1990) 3-26.

[6] Floyd, S. “TCP and Explicit Congestion Notification,” *ACM Computer Communications Review* 24 (1994) 10-23.

[7] Floyd, S., and Kevin Fall, “Promoting the Use of End-to-End Congestion Control in the Internet,” *IEEE/ACM Transactions on Networking* (1998).

[8] Floyd, S., and V. Jacobson. “Random Early Detection (RED) Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking* 1 (1993) 397-413.

[9] Gupta, A. et al. “Streaming the Digital Economy: How to Avert a Tragedy of the Commons,” *IEEE Internet Computing* 1 (1997) 38-46.

[10] Gupta, A., D. O. Stahl, and A. B. Whinston, “A Stochastic Equilibrium Model of Internet Pricing,” *Journal of Economic Dynamics and Control* 21 (1997) 697-722.

[11] Gupta, A., D. O. Stahl, and A. B. Whinston. “The Economics of Network Management,” *Communications of the ACM* 42 (1999) 57-63.

[12] Jacobson, V., and M. J. Karels. “Congestion Avoidance and Control,” *SIGCOMM’88,* (1988).

[13] Kalyanaroman, S., T. Ravichandran and R. Norsworthy. “Dynamic Capacity Contracting: A Framework for Pricing the Differentiated Services Internet,” (1998), http://www.ecse.rpi.edu/Homepages/shivkuma/research/papers/icepaper.html.

[14] Kleinrock, L. *Queueing Systems,* Vol. I & II, John Wiley & Sons, 1975.

[15] Kosiur, D. *Building and Managing Virtual Private Netowrks,* John Wiley & Sons, 1998.

[16] Lazar, A. A., and Nemo Semret. “Design, Analysis and Simulation of the Progressive Second Price Auction for Network Bandwidth,” *8th International Symposium on Dynamic Games and Applications* (1998).

[17] Li, Mingzhi et al. “Estimating Internet Users’ Demand Characteristics,” forthcoming in *Journal of Computational Economics* (2000), http://crec.bus.utexas.edu/works/articles/demand.pdf.

[18] Lin, Z. et al. “Exploring Traffic Pricing for the Virtual Private Network,” *WITS’99*, (1999).

[19] Lin, Z., D. O. Stahl, and A. B. Whinston. “A Traffic-pricing Model for the Packet-switching Network with Prioritized Round-robin Queueing,” *INFORMS-CIST’00*, (2000).

[20] MacKie-Mason, J., and H. Varian. “Pricing the Internet in Public Access to the Internet,” In B. Kahin and J. Keller (eds.), Englewood Cliffs, NJ:Prentice-Hall, 1995.

[21] Nagle, J. “Congestion Control in TCP/IP Internetworks,” Internet RFC 896 (1984), http://www.ietf.cnri.reston.va.us/rfc /rfc896.txt.

[22] Odlyzko, A. M. “A Modest Proposal for Preventing Internet Congestion,” *DIMACS Technical Report 97-68* (1997), http://www.dimacs.rutgers.edu/TechnicalReports/.

[23] Oppliger, R. *Internet and Intranet Security,* Artech House, 1998.

[24] Paxson, V., and S. Floyd. “Wide-Area Traffic: The Failure of Poisson Modeling,” *IEEE/ACM Transactions on Networking* 3(3) (1995).

[25] Rigney, C. et al. “Remote Authentication Dial In User Service (RADIUS),” Internet-draft (2000), http://www.ietf.org/internet-drafts/draft-ietf-radius-radius-v2-06.txt.

[26] Rigney, C., and Livingston. “RADIUS Accounting,” Internet RFC 2139 (1997), http://www.ietf.org/rfc/rfc2139.txt.

[27] Shenker, S. et al. “Pricing in Computer Networks: Reshaping the Research Agenda,” *Telecommunication Policy* 20 (1996) 183-201.

[28] Stahl, D. O., and A. B. Whinston. “A General Economic Equilibrium Model of Distributed Computing,” in W. W. Cooper and A. B. Whinston (eds.), *New Decisions in Computational Economics* (1994) 175-189.

[29] Stevens, W. “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,” Internet RFC2001 (1997), http://www.ietf.cnri.reston.va.us/rfc/rfc2001.txt.

[30] Tang, P., and C. Tai. “Network Traffic Characterization Using Token Bucket Model,” *Proceedings of IEEE InfoCom 99,* Vol.1 (1999).

[31] Thompson, K., G. J. Miller, and R. Wilder. “Wide-Area Internet Traffic Pattern and Characteristics,” *IEEE/ACM Transactions on Networking* (1997) 10-23.

Appendix : The Traffic-Pricing Formula for Round-robin Bandwidth Scheduling

In a network with round-robin queueing, a network node services all requests for the transmission through a channel by assigning each job a fixed-length time slice to use a full bandwidth. Each job is serviced in turn with a portion of the channel's bandwidth in the time slice. Newly-arrived jobs join the tail of the queue and will obtain its first time slice after the earlier arrivals have been serviced one round. We assume that the job size is generally distributed, the job arrival is a Poisson process and the bandwidth of the transmission channel is deterministic. These conditions turn the queueing system into a M/G/1 type. The deterministic bandwidth of the channel for the transmission results in each job having the same bandwidth share as others. In other words, a job's available bandwidth is inversely proportional to the number of jobs in the queue at the moment. We are to derive the formula for: 1) the expected number of jobs in the system, 2) a job's expected throughput time, and 3) the optimal pricing formula for a round-robin scheduling system. In default, all time measurements are based on the number of time slices.

Notations:

q: the job size in units of CPU time slices needed to process the job

B: the bandwidth of the network node in units of time slices per second

φ_j: the number of j-type jobs arriving per second

q_j: the size of j-type job in units of time slices

τ_j: job j's expected throughput time at the node

$\varphi = \Sigma_j \varphi_j$: the total job arrival rate

$E[q] = \frac{1}{\varphi}\sum_j \varphi_j q_j$: the average queue length

$E[q^2] = \frac{1}{\varphi}\sum_j \varphi_j q_j^{\,2}$: job rate weighted size-square

$p_j = \varphi_j q_j / \Sigma_j \varphi_j q_j = \varphi_j q_j / \varphi E[q]$: the proportion of job j in the system

$\rho = \Sigma_j \varphi_j q_j / B = \varphi E[q] / B$: bandwidth utilization ratio

L: a random variable for the number of jobs in the system

$L^* = E[L]$: the expected number of jobs in the system

<u>The Expected Number of Jobs in the System</u>

1) Let $E[s_q]$ be the expected service time of a job in the queueing system conditionally on a non-empty queue.

$$E[s_q] = \frac{1}{B}\sum_j p_j (q_j + 1)/2 = \frac{1}{2B}[1 + \sum_j p_j q_j]$$

$$= \frac{1}{2B}[1 + \sum_j \varphi_j q_j^{\,2} / \varphi E[q]] = \frac{1}{2B}\{1 + E[q^2]/E[q]\}$$

2) Since $E[L \mid L < 1] = 0$, we have

$$L^* = \rho E[L \mid L \geq 1] + (1 - \rho) E[L \mid L < 1] = \rho E[L \mid L \geq 1],$$

or $$E[L \mid L \geq 1] = L^* / \rho \qquad (a.1)$$

3) Let V denote the steady-state workload for the service in units of time slices

$$V = \rho \{E[s_q]\ (E[L \mid L \geq 1] - 1) + 1 / (2B)\} = E[s_q] *(L^* - \rho) + \rho / (2B)$$

Solving for L^*:

$$L^* = (2BV - \rho) / (2B\, E[s_q]) + \rho$$

But the workload of a M/G/1 queueing system is usually expressed in another form:

$$V = \rho[\frac{\rho E[q^2]}{2B(1-\rho)E[q]} + \frac{1}{2B}]$$

Hence[8],

$$L^* = \frac{\rho^2 E[q^2]}{(1-\rho)(E[q]+E[q^2])} + \rho \qquad (a.2)$$

In terms of φ:

$$L^*(\varphi) = \frac{(\sum_j \varphi_j q_j)^2}{B(B-\sum_j \varphi_j q_j)} \frac{\sum_j \varphi_j q_j{}^2}{(\sum_j \varphi_j q_j + \sum_j \varphi_j q_j{}^2)} + \frac{\sum_j \varphi_j q_j}{B}$$

Then

$$\frac{\partial L^*(\varphi)}{\partial \varphi_k} = \frac{q_k}{B}\{1 + \frac{\rho}{(1-\rho)(E[q]+E[q^2])}(\frac{(2-\rho)}{(1-\rho)}E[q^2] + \frac{B\rho[E[q]q_k - E[q^2]]}{E[q]+E[q^2]})\}$$

By conjecture, the last term $\frac{B\rho[E[q]q_k - E[q^2]]}{E[q]+E[q^2]}$ is negligible[9]. Therefore[10],

$$\frac{\partial L^*(\varphi)}{\partial \varphi_k} = (1 + \frac{\rho(2-\rho)E[q^2]}{(1-\rho)^2(E[q]+E[q^2])})\frac{q_k}{B} \qquad (a.3)$$

---

[8] Kleinrock [15] derived a simpler form $L^* = \frac{\rho}{1-\rho}$. We can prove that this is equivalent to (a.2) if E[q] << E[q^2].

[9] We ran an experiment with the pricing formula having this term and also let the simulation system calculate the price at the same time without the term. The experiment used the same experiment parameter values previously used by the GSW model simulation. The difference of the two prices is less than 0.1%.

[10] An approximate expression of (a.3) is $\frac{\partial L^*(\varphi)}{\partial \varphi_k} = \frac{\mathbf{1}}{(\mathbf{1}-\rho)^{\mathbf{2}}} \frac{q_k}{B}$ if E[q] << E[q^2]. It is the same as the one derived from Kleinrock's expression for the expected number of jobs.

The Expected Throughput Time

The expected throughput time for job *j* can be expressed as the summation of its service time and its waiting time in the queue:

$$\tau_j = q_j / B + \rho[1 / 2B + (E[L|L \geq 1] - 1) / B + E[L|L \geq 1]\ (q_j - 1) / B]$$

Substitute (a.1) into the above equation, we obtain:

$$\tau_j = [(L^* + 1)\ q_j - \rho / 2] / B \qquad (a.4)$$

Therefore,

$$\frac{\partial \tau_j}{\partial \varphi_k} = \frac{q_j}{B} \frac{\partial L^*}{\partial \varphi_k} - \frac{q_k}{2B^2} = (1 + \frac{\rho(2-\rho)E[q^2]}{(1-\rho)^2 (E[q]+E[q^2])})\ \frac{q_k q_j}{B^2} - \frac{q_k}{2B^2}$$

Generally, $q_j >> 1$, then $\frac{q_k q_j}{B^2} >> \frac{q_k}{2B^2}$. We have

$$\frac{\partial \tau_j}{\partial \varphi_k} \approx (1 + \frac{\rho(2-\rho)E[q^2]}{(1-\rho)^2 (E[q]+E[q^2])})\ \frac{q_k q_j}{B^2}$$

Define unit throughput time $\tau = \tau_j / q_j$.

$$\frac{\partial \tau}{\partial \varphi_k} = (1 + \frac{\rho(2-\rho)E[q^2]}{(1-\rho)^2 (E[q]+E[q^2])})\ \frac{q_k}{B^2} \qquad (a.5)$$

Substitute the L^* expression to *(b.5)*,

$$\frac{\partial \tau}{\partial \varphi_k} = (L^* + \frac{L^* - \rho}{1-\rho})\ \frac{q_k}{\rho B^2} \qquad (a.6)$$

Optimal Bandwidth Price Per Time Slice

Substitute (a.5) to the optimal price expression:

$$r_k = \sum_j \varphi_j \delta_j \frac{\partial \tau_j}{\partial \varphi_k}$$

$$\approx \frac{q_k}{\rho B^2}\,(L^* + \frac{L^* - \rho}{1-\rho})\,\sum_j \varphi_j \delta_j\, q_j$$

$$= \,(L^* + \frac{L^* - \rho}{1-\rho})\;q_k\;\bar{\delta}\,/\,B \qquad (a.7)$$

where $\sum_j \varphi_j q_j \delta_j = \bar{\delta}\,\rho B$. $\bar{\delta} = \dfrac{\sum_j \varphi_j q_j \delta_j}{\sum_j \varphi_j q_j}$ is the mean of δ_j weighted by data volume rates $\varphi_j q_j$.

11

The unit price is

$$r = r_k / q_k = \,(L^* + \frac{L^* - \rho}{1-\rho})\;\bar{\delta}\;/\,B \qquad (a.8)$$

(a.8) implies that the price is linear in terms of job size.

[11] The accurate expression for r_k contains two more terms. One of them is in a quadratic form and hence is inconvenient in use. We ran a set of experiments to compare the outcomes from the approximate formula with the exact expression. The results show that the difference in price level is within 0.1%. This level of error has in fact no effect on the accuracy of the outcomes because another group of experiments show that the prototype VPN pricing system can tolerate a price bias up to 5% without a significant impact on the welfare rate.