

Pricing Network Services

Jun Shu

Department of Industrial Engineering
and Operations Research
UC Berkeley
Email: jshu@ieor.berkeley.edu.

Pravin Varaiya

Department of Electrical Engineering
and Computer Science
UC Berkeley
Email: varaiya@eecs.berkeley.edu.

Abstract— We propose a game theoretic pricing mechanism for statistically guaranteed service in packet-switched networks. The mechanism provides congestion control, differentiated qualities of service, and efficient resource allocation. For users, the mechanism offers better quality and lower price. Service providers can base new service and revenue models within the mechanism. We apply this mechanism to the Internet.

I. INTRODUCTION

The predominant form of pricing of Internet services in the United States is flat rate pricing. Residential users subscribe to certain amount of bandwidth to access the Internet at a monthly flat fee. Businesses may use customized contracts, almost all of which are also flat rate based. The flat rate tariff grants a subscriber unlimited use of the network. A flat rate pricing scheme encourages waste, increases cost, and forces light users to subsidize heavy users according to experiments at Berkeley [4], [24]. An inevitable consequence is the emergence of negative congestion externalities.

For service providers, flat rate increases per subscriber recruitment and retention cost, and lowers service quality. In fact the providers cannot generate sufficient revenue from the network service to sustain the enormous investment required to provide the service. The lack of a proper pricing and revenue model is partly responsible for the current difficulties of the telecommunication industry. Bundling seems to be the only way to survive. Witness the merger between American-Online and Time-Warner—a marriage between network service and content. The jury is still out on whether this is a viable alternative.

At the same time, network engineers have developed approaches to deliver quality of service (QoS). The Internet generally provides a single quality of service known as the “best effort” datagram delivery. This service model is scalable and robust, and can support different network applications provided that there is no congestion [6]. Different applications require different levels of guarantees of QoS, measured in terms of delay, jitter, and loss. When there is congestion, packets may be dropped and delayed, and delivery guarantee becomes variable.

A core issue of concern to network users, service providers, and engineers is the constantly changing network behavior that is under nobody’s control. The network behavior depends on the aggregated traffic load of the network—the result of many users’ individual decisions on how to use the network. These

decisions are affected by the incentives users face. Thus, our approach is to bring price into network engineering design as a signal of network control. In doing so, we address both the engineering problem of delivering QoS and the economic problem of pricing network services.

Our work is inspired by MacKie-Mason and Varian who in 1994 proposed a “smart market” mechanism [19] that suggests an auction based scheme to price congestion. The smart market mechanism remains a preliminary proposal. Our mechanism, named the Smart Pay Admission Control (SPAC) mechanism, provides QoS differentiation in addition to congestion control. This feature makes SPAC practical since flows of packets can be treated as aggregates of different levels of QoS, a well researched approach in networking, e.g., DiffServ [21].

The rest of the paper is organized as follows. In section II, we illustrate the relevance of incentives for network congestion control. Section III presents the SPAC mechanism. We apply the mechanism to a pricing scheme for a network architecture in section IV. Section V concludes the paper with a summary.

II. THE LACK OF INCENTIVES FOR CONGESTION CONTROL

The Internet provides congestion control through the Transport Control Protocol (TCP) [13]. Congestion is inferred from the absence of acknowledgements of packets from the sender. When there is no congestion, the sender slowly but continuously increases the rate of sending packets. As soon as congestion occurs, the sender halves the sending rate. This “multiplicative decrease and additive increase” is the key to the TCP algorithm.

If everyone uses TCP, congestion could be managed. However, there is one problem: there is no intrinsic incentive for a user to submit to the congestion control algorithm. We illustrate this point from a game theoretic perspective. We start with a simple model of two users, and then analyze a general model of an arbitrary number of users.

A. Two-User Case

Consider a hypothetical scenario with only two users (User A and User B) sending packets, at rate 1, through a bottleneck. The bottleneck can allow one packet to pass through in each time interval. When two packets compete, the bottleneck randomly admits one packet to pass and discards the other. We model how the two users might interact in a game of strategic

form. The game is played over a TCP acknowledgement timeout interval for a packet waiting to be sent in the next time interval.

Each user may choose one of two opposing strategies. One strategy is to follow an authentic TCP congestion control scheme which decides not to send the packet because the timeout has occurred. We call this strategy “follow-TCP.” The other strategy sends the packet despite the timeout signal. We call this strategy “cheat-TCP,” because it represents behavior that amounts to cheating in a community adhering to TCP.

The payoff to each user is the probability of a packet getting through the bottleneck in the next time interval. Obviously, if the packet is not sent at all, the payoff is zero. If the packet is sent when there is no other packet in the bottleneck, the payoff will be one. When two packets are competing for the bottleneck, we assume that the probability of each packet getting through is p_A and p_B for user A and B respectively, where $p_A > 0$, $p_B > 0$, and $p_A + p_B = 1$.

The bi-matrix in Table I summarizes the different payoffs to the two users under all possible outcomes of the game. Each cell of the bi-matrix first lists the payoff to the row player, User A, and then the payoff to the column player, User B. For example, when both users choose follow-TCP strategy, the payoffs are 0 to both users (the upper-left cell) because no packets are sent. If A chooses follow-TCP and B chooses cheat-TCP then A receives payoff 0 and B receives payoff 1 (the upper-right cell). When both users choose cheat-TCP strategy, the payoffs are p_A and p_B (the lower-right cell).

TABLE I
CONGESTION IN A TWO-USER GAME

		User B	
		follow-TCP	cheat-TCP
User A	follow-TCP	0, 0	0, 1
	cheat-TCP	1, 0	p_A, p_B

Evidently User A is always better-off choosing cheat-TCP regardless of what User B does. So is User B by choosing the same cheat-TCP strategy regardless of what User A does. Therefore, the strategy profile in which both users choose cheat-TCP is a dominant strategy equilibrium of this game. Furthermore, in any other combination of strategies, at least one user has an incentive to deviate from that combination. Clearly, it is not a stable state for both users to choose the follow-TCP strategy.

Although our two-user game model is highly simplified, it demonstrates the lack of incentives in TCP congestion control.

B. Many-User Case

Now consider a packet-switched network with n users. Let r_i denote user i 's rate of sending packets, $i = \{1, \dots, n\}$. The total rate, $L = \sum_{i=1}^n r_i$, is a measure of the network load. The maximum rate the network can accommodate is L_{\max} . Suppose that

- 1) rates are continuously divisible;

- 2) users' cost to send packets is a charge of c per unit rate¹; and
- 3) each user's value of sending packets is a function of the network load, $v(L)$ per unit rate; and the marginal value decreases as the network load increases, i.e., $v(L) > 0$ for $L < L_{\max}$, $v(L) = 0$ for $L \geq L_{\max}$, $v'(L) < 0$ and $v''(L) < 0$ for $L < L_{\max}$.

A central planner, acting on behalf of all the users, would find the optimal solution to

$$\max_{0 \leq L < \infty} Lv(L) - Lc, \quad (1)$$

the first-order condition for which is

$$v(\hat{L}) + \hat{L}v'(\hat{L}) - c = 0 \quad (2)$$

where \hat{L} is the optimal network load.

However, in reality, there is no central planner in a decentralized network such as the Internet. The users simultaneously choose how fast to send packets based on their private incentives. We model this feature as a strategic game.

A strategy for user i is a sending rate, $r_i \in [0, \infty)$. The payoff to user i from sending at r_i is

$$r_i[v(L) - c]. \quad (3)$$

Let (r_1^*, \dots, r_n^*) , a profile of rates, one for each user, be a Nash equilibrium. Then, for each user i , r_i^* must maximize the payoff function (3) given that the other users choose a profile of rates, $(r_1^*, \dots, r_{i-1}^*, r_{i+1}^*, \dots, r_n^*)$. Let $L_{-i}^* = \sum_{j=1, j \neq i}^n r_j^*$. The first-order condition for maximizing the payoff (3) is

$$v(r_i + L_{-i}^*) + r_i v'(r_i + L_{-i}^*) - c = 0$$

Summing over all n users' first-order conditions yields

$$v(L^*) + \frac{1}{n} L^* v'(L^*) - c = 0 \quad (4)$$

where L^* is the network load under the Nash equilibrium, $L^* = \sum_{i=1}^n r_i^*$.

It is straightforward to prove that $L^* > \hat{L}$, that is, too many packets (L^*) are sent in a Nash equilibrium compared to the optimal level of network load, \hat{L} . The network resource is over-utilized because each user considers only her own incentives, not the effect of her actions on the other users. This is how a “successful” network—in terms of user popularity—can run into serious trouble without proper incentives for usage and congestion control. Economists and political philosophers have long recognized this phenomenon as “the tragedy of commons” [11].

¹Instead of *per unit rate* charge, one could also assume a charge of *flat fee*. The result of the analysis is the same.

C. Information Asymmetry

Interestingly, despite the lack of incentive to conform to it, the TCP congestion control algorithm seems to be working fine in today's Internet. Our explanation is "information asymmetry." TCP has been widely adopted by all major software companies and works transparently to end users. Few people are aware of the incentive. Few people know how to change their software to gain in this game of congestion control.

However, this lack of knowledge does not mean the problem does not exist. It is merely a good fortune, which depends upon a few "benevolent" dictators (i.e., organizations that implement TCP/IP stacks). The knowledge will spread; and once widely known, users will "cheat."

In fact, a number of applications bypass the TCP congestion control scheme. They include FlashGet, GolZilla, ReGet, Download Accelerator, GetRight, GetSmart, and Download Devil, as noted in [14]. These applications parallelize the download of each web object by opening multiple connections per object and downloading a different portion of the object on each connection. Some commercial versions of these applications open additional connections during congestion. Although each connection still uses TCP, these applications behave in a much more aggressive way that in effect abandon the TCP congestion control without changing it.

Moreover, modifying the source code of TCP implementation is not difficult. The number of lines of code involved can be as few as a dozen (for FreeBSD). With the popularity of open source software such as Linux, one can imagine the time when the grandma next door starts to cheat TCP by using a plug-in downloaded by her granddaughter.

III. THE SMART PAY ADMISSION CONTROL MECHANISM

Since congestion is caused by too many users competing for a limited resource, our objective is to find an economically efficient way to allocate network resource among users. In a packet-switched network, network resource usage is reflected in the statistical guarantee the packets receive. This means that the higher the value a user puts on the service (packet delivery), the better statistical guarantee of delivery the user should get, especially during congestion. This is our notion of "congestion control" and "economically efficient resource allocation." The different guarantees of delivery form the basis of "differentiated qualities of service."

However, this goal requires that we know the *true value* each user puts on the network service. We designed a mechanism, the Smart Pay Admission Control (SPAC) mechanism, in which every user has an incentive to voluntarily disclose this "true value" out of his or her own selfish concerns.

This section presents the SPAC mechanism. First, we illustrate the main idea behind the mechanism (III-A). Then, we lay down some formal definitions (III-B) and describe the actual admission control algorithm (III-C). Next, we analyze why users would want to disclose their true values (III-D). The complete proof is in Appendix A. Finally, we show the conditions under which users and service providers would

want to participate in this mechanism (III-E, Appendix B, and III-F).

A. An Illustration of the Main Idea

The SPAC mechanism is an auction-based admission control algorithm.

A service provider sells a service to a group of users. The service has a range of different qualities: QoS, defined as the statistical guarantee of the service delivery, the probability that the service will be completed successfully. The higher the probability, the better is the quality.

Users bid for the service by announcing how much they are willing to pay for the service, and in return, receive admission tickets (to the service) in different colors. These different colors represent different qualities of service. Every user is served with certain quality according to the color of her ticket. Notice that although users are aware of the differentiated QoS, they announce their value of the service, not the values of different qualities.

The SPAC mechanism decides which color (QoS level) each user is entitled to, based on the bids of all users. For instance, suppose that the service has three different levels of QoS: level- 2, 1 and 0 (level-2 being the highest quality), and each level can accommodate N_2 , N_1 and N_0 number of users respectively. All bids are sorted (equal bids are ordered randomly among themselves). The highest N_2 bidders receive the service at the highest quality, level-2. The next highest N_1 bidders get quality level-1, and the remaining bidders get the lowest quality, level-0.

The SPAC mechanism requires that the lowest quality level accommodate *all* remaining users who are rejected for any higher quality levels. This may be accomplished by arbitrarily lowering the quality of the lowest QoS level. This means that every user is admitted to the service. Even if a user bids zero (bidding negative values is not allowed), she still receives the service, albeit the quality may be the lowest, or, may be not, depending how other users bid. We call this feature the "universal coverage" constraint.

The users must pay a price, known as the congestion fee, for the QoS they receive. For the lowest level QoS, the congestion fee is *always* zero. This is analogous to the current free "best effort" service in the Internet. For other higher level QoS, the fee is calculated based on a variant of the generalized Vickrey auction, also known as the VCG (Vickrey-Clark-Groves) mechanism [25], [1], [10]. The actual formula for the price is presented in III-B.

Based on the payment scheme in SPAC, it can be shown that the best bid for each user is to announce her *true value* of the service, regardless of what other users may do (Proposition 1 in III-D). Thus, the SPAC mechanism achieves the economically efficient allocation of different qualities among a group of autonomous and selfish users.

Furthermore, both users and the service provider have incentives to participate in this mechanism. For the users, they are assured that if they are served at the lowest level of QoS, they will pay nothing. This is exactly what they are

getting now in the current Internet.² If they are willing to pay more, they can receive better service, but they will not be overcharged above their willingness to pay—the value they obtain from the successful delivery of the service is always greater than the charge they have to pay (Proposition 2 in III-E).

The service provider can devise the network resource to provide a variety of services and is rewarded with extra compensation for the services. There is no longer undersupply nor oversupply of the resource, users practically put themselves into the right categories of services [17].

B. The Formal Definition

Formally, a *mechanism* is a game with players, outcomes, players' strategies, outcome functions and players' payoff functions. We define the SPAC mechanism, $\mathcal{M}_{\text{SPAC}}$, as follows.

1) *Players*: There are $n + 1$ players including n agents, denoted by player $i = 1, \dots, n$, and, one principal, denoted by player $i = 0$.

2) *Profile of Values*: Borrowing a common notation from game theory, we refer to a collection of values of some variable (e.g. α), one for each player, as a *profile*, denoted by $(\alpha_1, \dots, \alpha_n)$. For any profile $\alpha = (\alpha_1, \dots, \alpha_n)$ and any $i \in n$, let α_{-i} be the list of elements of the profile α for all players except i , that is, $\alpha_{-i} = (\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n)$.

3) *Service and QoS*: The principal provides a service to the agents. The principal statistically guarantees the service at different delivery rates—the probability of successfully completing the service. The QoS (quality-of-service) is defined as the delivery rate of the service.

There are m levels of different QoS, denoted by $k = 0, 1, \dots, m - 1$. Let d_k denote the service delivery rate of k -th level, and $d = (d_0, d_1, \dots, d_{m-1})$ where $0 \leq d_0 < d_1 < \dots < d_{m-1} < 1$.

The provisioning of the service consumes resource. Given resource capacity \mathcal{C} , the principal allocates \mathcal{C} among m levels so that each level can admit at most $A_k(\mathcal{C})$ number of agents. When fixed as a constant, $A_k(\mathcal{C})$ is simply denoted by A_k .

4) *Agents' Actions*: An agent receives the service at QoS level $k = 0, 1, \dots, m - 1$. This level of QoS is decided by the principal based on how all of the agents value the service.

Prior to receiving the service, each agent i is expected to announce her value of the service. The *announced values*, also called *bids*, are denoted by $b = (b_1, \dots, b_i, \dots, b_n)$ where b_i is the bid for agent i . The bids are disclosed at least to the principal.³ The *true values*, also referred to as *types*, which are the agents' private information, are denoted by $\theta = (\theta_1, \dots, \theta_i, \dots, \theta_n)$ where θ_i is the true value of agent i . Let \mathcal{B}_i denote the space of allowable bids and Θ_i the space of agent type for agent i , $i \in \{1, \dots, n\}$.

Let $b_{(1)}, b_{(2)}, \dots, b_{(n)}$ be the order statistics corresponding to b_1, b_2, \dots, b_n . That is, $b_{(i)}$ is the i th smallest value among b_1, b_2, \dots, b_n .

²In the current Internet, a flat fee is still charged. However, there is no usage-based charge.

³In reality, bids are most likely disclosed to the principal only. In our formal analysis, it does not make a difference if the bids are disclosed to other players.

After receiving the service, each agent pays a congestion fee (defined later) calculated by the principal.

5) *Principal's Action*: Based on agents' bids, the principal decides at what QoS level each agent should be served. Formally, the principal computes a solution, a vector $x = (x_1, \dots, x_i, \dots, x_n)$, in which x_i is the service delivery rate that agent i is receiving. Thus $x_i \in X$ and $X = \{d_0, d_1, \dots, d_{m-1}\}$. Let the function $Q_k(x_i)$ indicate the level of QoS agent i receives,

$$Q_k(x_i) = \begin{cases} 1 & \text{if } x_i = d_k \\ 0 & \text{otherwise} \end{cases}$$

and for all $i \in \{1, \dots, n\}$, each agent i is served at one and only one QoS level:

$$\sum_{k=0}^{m-1} Q_k(x_i) = 1.$$

The solution $x^*(b)$, a function of b , is efficient (i.e., maximizing the total benefit to all users) and feasible if

$$x^*(b) = \arg \max_{x_i \in X} \sum_{i=1}^n b_i x_i \quad (5)$$

subject to the capacity constraint: $\forall k = 1, \dots, m - 1$,

$$\sum_{i=1}^n Q_k(x_i) \leq A_k \quad (6)$$

and subject to the universal service coverage (i.e., everyone is admitted) constraint

$$\sum_{k=0}^{m-1} A_k \geq n. \quad (7)$$

6) *Congestion Price*: Based on agents' bids, the principal computes a congestion price for each level of QoS. Let $p = (p_0, p_1, \dots, p_{m-1})$ denote all prices, where the price for level-0 is a constant zero,

$$p_0(b) \equiv 0 \quad (8)$$

and the price for level- k is, $\forall k = 1, \dots, m - 1$,

$$p_k(b) = p_{k-1}(b) + (d_k - d_{k-1})b_{(n - \sum_{l=k}^{m-1} A_l)}. \quad (9)$$

See III-C for an interpretation of $b_{(n - \sum_{l=k}^{m-1} A_l)}$.

7) *Agents' Utility*: For all $i \in \{1, \dots, n\}$, agent i 's utility (payoff) function is

$$u_i(b, \theta_i) = \theta_i x_i - \sum_{k=0}^{m-1} p_k Q_k(x_i) \quad (10)$$

where $\sum_{k=0}^{m-1} p_k Q_k(x_i)$ is agent i 's payment for receiving the service at certain QoS level.

C. Solutions to Principal's Problem

Based on agents' bids, the principal must decide how to provide the service to all the agents at different levels of QoS. If we fix the number of agents the network can serve at each level of QoS (excluding the lowest level), that is $\{A_k : k = 1, \dots, m-1\}$, then the Principal's problem becomes an admission control problem—the principal must decide which agents to admit at each level of QoS.

This admission problem is formulated in (5), (6), and (7). The solution is straightforward. First sort all the bids in descending order. The highest A_{m-1} bidders are admitted to the QoS level d_{m-1} , the next higher A_{m-2} bidders to level d_{m-2} , and so on, until all the bidders are admitted to some QoS levels.

Calculation of the congestion prices as defined in (8) and (9) is straightforward after sorting all the bids. Notice that according to the admission control algorithm, $b_{(n-\sum_{l=k}^{m-1} A_l)}$ is the highest bid of agents who are rejected from the k -th or above levels of QoS.

D. Agents' Strategies

A strategy for agent i is how much she should bid, $b_i \in [0, \infty)$. A rational agent chooses a strategy that will maximize the agent's utility, which is dependent on the actions taken by all agents.

However, we claim that for the “linear” utility function (10), the optimal strategy for every agent, regardless of other agents' strategies, is to announce her true value. That is, $b_i = \theta_i$ is the solution to maximize (10) for all $i \in \{1, \dots, n\}$.

A strategy is *dominant* for an agent if it is optimal for the agent regardless of other agents' strategies. A mechanism is *dominant strategy incentive compatible* or *strategy-proof* if it is a dominant strategy for every agent to announce her true value (or type).

PROPOSITION 1: The Smart Pay Admission Control mechanism, $\mathcal{M}_{\text{SPAC}}$, is dominant strategy incentive compatible (or strategy-proof).

We include the complete proof in Appendix A. In fact, the SPAC mechanism is a special case of the Clarke-Groves mechanism [1], [10] and can be viewed as a generalized Vickrey auction [25].

E. Agents' Participation

Why would a user want to participate in the SPAC mechanism in the first place? Since agents' participation in $\mathcal{M}_{\text{SPAC}}$ is voluntary, $\mathcal{M}_{\text{SPAC}}$ must provide incentives for the agents to participate.

First, $\mathcal{M}_{\text{SPAC}}$ must guarantee that agents will not be overcharged.

PROPOSITION 2: In the Smart Pay Admission Control mechanism, $\mathcal{M}_{\text{SPAC}}$, every agent's utility (payoff) from the mechanism is nonnegative.

The proof is in Appendix B. This property states that the utility function (10) is always nonnegative for all agents. In other words, agents can rest assured that the benefit they obtain

from the service is always greater than or equal to the charge they have to pay.

Second, an agent would participate in a mechanism only if her expected payoff from the participation is at least as large as that from *not* participating in the mechanism. This condition is known as the *participation constraint* or *individual rationality*.

Let $\bar{u}_i(\theta_i)$ denote the payoff function that agent i can receive by withdrawing from $\mathcal{M}_{\text{SPAC}}$ when her type is θ_i . We can define three types of participation constraints for each agent: $\forall i \in \{1, \dots, n\}$,

- *ex ante* participation constraint:

$$E_{b_i}[E_{b_{-i}}[u_i((b_i, b_{-i}), \theta_i)|b_i]] \geq E_{\theta_i}[\bar{u}_i(\theta_i)]; \quad (11)$$

- *interim* participation constraint:

$$E_{b_{-i}}[u_i((b_i, b_{-i}), \theta_i)|b_i] \geq \bar{u}_i(\theta_i); \quad (12)$$

- *ex post* participation constraint:

$$u_i((\theta_i, \theta_{-i}), \theta_i) \geq \bar{u}_i(\theta_i). \quad (13)$$

In (11) and (12) E denotes expectation.

In cases where agent i is only allowed to refuse to participate before the agents learn their types, $\mathcal{M}_{\text{SPAC}}$ must satisfy the *ex ante* participation constraints in order to attract agents' participation. In other cases, if agent i is allowed to withdraw from the mechanism after agents have learned their types and before they have chosen their actions, then $\mathcal{M}_{\text{SPAC}}$ must satisfy the *interim* participation constraints to convince the agents to stay in the mechanism. In still other cases, if there is no way to bind the agents to the assigned outcomes of $\mathcal{M}_{\text{SPAC}}$ against their will, i.e., agent i can withdraw at any time, then to insure agent i 's participation, $\mathcal{M}_{\text{SPAC}}$ must satisfy the *ex post* participation constraints.

We now discuss how the principal should devise the service to satisfy these constraints so as to ensure the agents' voluntary participation.

Suppose that

$$\bar{u}_i(\theta_i) = \bar{d}\theta_i \quad \forall i \in \{1, \dots, n\} \quad (14)$$

where \bar{d} is the service delivery rate agent i can receive without participating in $\mathcal{M}_{\text{SPAC}}$. The conditions under which the participation constraints can be satisfied depend on two things:

- the difference of the service delivery rates between \bar{d} and $d = (d_0, d_1, \dots, d_{m-1})$, and
- the distributions of agent types, θ_i for all $i \in \{1, \dots, n\}$.

Given \bar{d} and the distributions of agents' types, the principal can adjust the service delivery rate vector d to satisfy different variety of the participation constraints defined in (11), (12) and (13). Based on the actual business plans, the principal decides which types of participation constraints should be satisfied.

Let $q, q \in \{0, \dots, m-1\}$, be a random variable that denotes the QoS level assigned to agent i . According to definition

(10)⁴, agent i 's payoff function from the mechanism is

$$u_i(b, \theta_i) = \theta_i x_i - \sum_{k=1}^q (d_k - d_{k-1}) b_{(n - \sum_{l=1}^{k-1} A_l)}.$$

If the distributions of agents' types are unknown, we can say little about q . However, the principal can still satisfy the ex post participation constraint (13) by setting $d_0 = \bar{d}$, providing a lower bound of payoff function that is equal to the utility to the agents should they withdraw. This is true because an agent is guaranteed QoS level-0 which provides the service at the delivery rate of d_0 and at no cost. The proof is a straightforward extension of the proof of Proposition 2.

If the distribution of agents' types is available (e.g. through market research and historical usage data), tighter lower bounds can be achieved for all three different participation constraints. Tighter lower bounds are attractive to the principal because they mean more efficient resource usage.

F. Principal's Participation

Why would a service provider adopt the SPAC mechanism in the first place? The rationality for the principal, the service provider, is similar to that of the agents' participation constraints. Compared with the current flat rate service model, $\mathcal{M}_{\text{SPAC}}$ produces better payoff to the principal. In particular, $\mathcal{M}_{\text{SPAC}}$ allows a service provider to execute price discrimination among users, collect the congestion fee, and provide better service satisfaction to users. New revenue models can be built based on the SPAC mechanism. Section IV demonstrates one such example.

IV. SPAC BASED PRICING FOR DIFFSERV

We now specify the SPAC mechanism for network service management and congestion control in the form of a pricing scheme for the DiffServ network architecture.

In the DiffServ architecture [21], traffic entering a network is classified, possibly conditioned at the boundaries of the network, and assigned to different aggregates. Each aggregate is identified by a specific DiffServ codepoint in the packet header. Within the core of the network, packets are forwarded at each node according to certain per-hop behavior associated with the DiffServ codepoint. Packets from different aggregates are treated differently; hence service differentiation is achieved. Scalability is achieved through three major features of the DiffServ model. First, service differentiation is given to packet aggregates rather than individual microflows. Second, traffic classification and conditioning processes are pushed to the edge of the network. Finally, service differentiation information is stored inside packet headers rather than network nodes so that the network does not need to maintain traffic states.

The service differentiation, conditioning and marking at the boundary allow for a smooth application of the SPAC mechanism, which is essentially an admission control algorithm. Our DiffServ pricing scheme consists of two parts: pricing for

traffic profiles and pricing for out-profile traffic streams. The SPAC mechanism is applied only to the second part, pricing for out-profile traffic.

A. Pricing for Traffic Profiles

A traffic profile (TP) is a description of the temporal properties of a traffic stream. A well-defined TP provides rules for determining whether a particular packet is in or out of profile. For example, a TP may specify a rate R and a burst size B . A token bucket meter with rate R and buffer size B can measure a traffic stream against this TP. A packet is *in-profile* if, when the packet arrives, there are sufficient tokens in the bucket; a packet is *out-profile* if there are insufficient tokens.

More sophisticated TPs can be devised to accommodate different types of traffic streams with different QoS requirements. For instance, Voice-over-IP (VoIP) can have a VoIP TP; video conferencing can have its own TP. Moreover, there can be a premium quality VoIP TP and a not-so-good quality VoIP TP. A TP may implicitly include certain MPLS commitment to ensure end-to-end QoS. The type and number of TPs will continue to grow as network applications evolve.

A traffic profile is sold at a flat fee for certain time period with unlimited usage. The in-profile traffic will incur no extra fee. For example, a user may purchase a VoIP TP for a month during which she can use it as much as she likes. The trading mechanism for traffic profiles is deliberated left undefined. A service provider may choose to directly sell traffic profiles to their customers—very much like how AOL sells a dialup access connection for \$25 per month. Or, a secondary market may exist to allow users to exchange their traffic profiles. The flat rate pricing of TP maintains a sense of monetary predictability for both users and service providers. The type and number of marketplaces for TPs will develop as the network service business evolves and matures.

B. Pricing for Out-Profile Traffic

The out-profile traffic is the potential congestion maker. While there is no limit on how much out-profile traffic a user may generate, the user must pay a congestion fee for the out-profile traffic. The congestion fee is calculated according to the SPAC mechanism, $\mathcal{M}_{\text{SPAC}}$, defined in section III. Following the terminology in DiffServ, a Bandwidth Broker (BB) assumes the task of congestion fee calculation.

More specifically, the $\mathcal{M}_{\text{SPAC}}$ -based congestion pricing works in the following fashion.

- 1) The service provider decides how many different levels of QoS the network will provide. This is the "service delivery rate" vector, $d = (d_0, d_1, \dots, d_{m-1})$, defined in $\mathcal{M}_{\text{SPAC}}$.
- 2) The different service delivery rates are statistically guaranteed through a Resource Manager that dynamically configures the interior of the network, setting queue size and bandwidth on routers for different classes of traffic. Each QoS level has its own DSCP (DiffServ Codepoint) identifying a PHB (Per Hop Behavior) aggregate. We

⁴See also Lemma 1 in Appendix A.

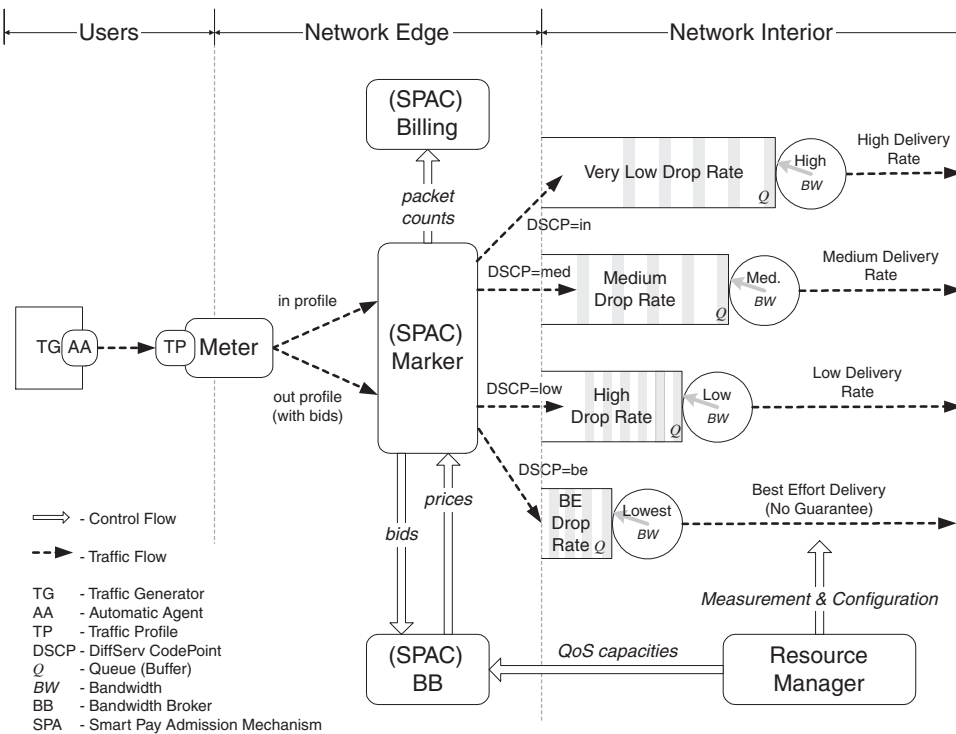


Fig. 1
SPAC-BASED PRICING FOR DIFFSERV

have developed a tool for this type of network management [23].

- 3) An Automatic Agent (AA), acting on behalf of the users, states (or bids) the true value (due to *incentive compatibility* of $\mathcal{M}_{\text{SPAC}}$) of service for the out-profile traffic.
- 4) A Bandwidth Broker (BB), based on all of users' bids, calculates the congestion fee for each QoS level, i.e., the same $p = (p_0, p_1, \dots, p_{m-1})$ as defined in $\mathcal{M}_{\text{SPAC}}$. BB receives from the Resource Manager current capacity information for different levels of QoS, i.e., the $\{A_k, k = 1, \dots, m-1\}$.
- 5) A Marker marks the DSCP of the out-profile traffic according to the QoS level assigned to the out-profile traffic. The assignment is according to the admission control algorithm defined in $\mathcal{M}_{\text{SPAC}}$.
- 6) A Billing system records the out-profile packet counts from the Marker, and the temporal congestion fee.

Fig. 1 is a schematic representation of the $\mathcal{M}_{\text{SPAC}}$ -based DiffServ pricing. The figure shows one traffic source (TG), one traffic profile (TP) and four different levels of QoS. The highest quality is reserved for in-profile packets marked as DSCP=in and provides a high rate of (successful) delivery. The other three QoS levels are for out-profile packets, marked as DSCP=med, DSCP=low, or DSCP=be for the medium, low, or best-effort rates of (successful) delivery, respectively.

To ensure users' participation in this scheme, as we have

discussed in section III-E, the lowest quality is that of the best-effort, which is the current quality of the Internet.

$\mathcal{M}_{\text{SPAC}}$ also ensures that all bids reflect the true value of service perceived by users, and that congestion is well controlled at all QoS levels above the best-effort level, because the correct amount of traffic is admitted for service at those levels.

Notice that the bandwidth broker, the meter, the marker, and the billing system all reside at the edge of the network. This feature, inherited from DiffServ, assures the scalability of this pricing scheme. The interior of the network needs not be aware of the SPAC mechanism.

The end-to-end delivery of packets is achieved through the bilateral agreements between the DiffServ domains along the delivery path. MPLS can also be used to provide better quality guarantee along certain paths. The issue of end-to-end QoS is addressed by network architecture, not the pricing scheme *per se*.

When it comes to inter-domain pricing, the network provider of one DS domain will aggregate its outgoing traffic and become the user of its downstream domain. The same pricing scheme can be applied.

V. SUMMARY AND CONCLUSION

We started this paper by recognizing the lack of pricing structure and service differentiation in the current Internet. We then showed, both intuitively and formally, that the fundamental problem lies in the incentives faced by individual

network users. The opportunity to solve the problem also centers upon incentives. As the major theme of the paper, we proposed the Smart Pay Admission Control mechanism. We proved the incentive compatibility and the individual rationality (the participation constraints) of this mechanism. Finally, we proposed a pricing scheme based on the SPAC mechanism within the DiffServ network architecture.

The basic idea of SPAC-based pricing is to mark packets according to their value to their senders. Due to the special design of the SPAC mechanism, senders have incentives to reveal their true values. For DiffServ, the marking occurs at the edge; packets are treated differently according to the marking, in the interior of the network. The higher the value of packets, the better the treatment they receive. One can also apply SPAC-based pricing to MPLS type networks in which traffic profiles contain routing information and thus congestion is reflected along a path instead of in a domain.

With our pricing scheme, congestion control is achieved in an economically efficient way—through the congestion fee which serves as a signal to users. Those users who value the service less will voluntarily back down when congestion occurs. The network load stays in an equilibrium through each user’s individual, autonomous, and selfish decisions on how to use the network.

Our pricing scheme allows a service provider to devise a comprehensive set of service plans with different QoS characteristics. The service provider should be able to offer more services at cheaper prices. In the meantime, the provider will be able to collect congestion fee. Both results will increase revenue while expanding the customer base.

To network users, our pricing scheme provides predictability through the concept of traffic profiles. With cheaper and more customized services, more users will be able to afford high speed broadband network services. In addition, users will have the flexibility of when to pay and not to pay for the service (as oppose to the current flat rate structure in which users have to pay no matter how they use the network).

Given a good pricing mechanism, the next challenge is to implement it. Unfortunately, the state of the art for managing IP networks involves manual configuration of each IP router, and traffic engineering based on limited measurements. The network industry is lacking in software systems that a service provider can use to support traffic measurement and dynamic configuration. We have developed a network management software toolkit to fulfill this need. We call our software system SNT. A technical paper on SNT is available to interested readers [23]. With SNT, our SPAC-based pricing scheme becomes implementable.

APPENDIX A PROOF OF PROPOSITION 1

We prove that $\mathcal{M}_{\text{SPAC}}$ is dominant strategy incentive compatible (or strategy-proof).

For any agent i , $i \in \{1, \dots, n\}$, let $x_{-i}^*(b_{-i})$ denote the

optimal admission outcome when agent i is absent. That is,

$$x_{-i}^*(b_{-i}) = \arg \max_{x_j \in X} \sum_{j=1, j \neq i}^n b_j x_j$$

subject to

$$\sum_{j=1, j \neq i}^n Q_k(x_j) \leq A_k \quad \forall k = 1, \dots, m-1 \quad (15)$$

$$\text{and} \quad \sum_{k=0}^{m-1} A_k \geq n-1.$$

Let $v_j(x, b_j)$ denote the (declared) value to agent j from the service received, which is a function of the admission outcome and the agent’s bid. That is,

$$v_j(x, b_j) = b_j x_j, \quad \forall j \in \{1, \dots, n\}. \quad (16)$$

First, we transform the payoff function defined in (10).

LEMMA 1: For all $i \in \{1, \dots, n\}$ there exists $q \in \{0, \dots, m-1\}$ such that $Q_q(x_i^*(b)) = 1$, and

$$\sum_{j=1, j \neq i}^n v_j(x_{-i}^*(b_{-i}), b_j) - \sum_{j=1, j \neq i}^n v_j(x^*(b), b_j) = p_q(b). \quad (17)$$

Proof: By (7), the universal service coverage constraint, every agent is admitted to one level of QoS, therefore there exists $q \in \{0, \dots, m-1\}$ such that $Q_q(x_i^*(b)) = 1$.

The left hand side of (17) is the impact of agent i ’s participation on all other agents j , $j \neq i$. Without the participation of agent i , all other agents would have been admitted to certain QoS levels according to the solution $x_{-i}^*(b_{-i})$. With agent i ’s participation, suppose that agent i is admitted to QoS level q according to the solution $x^*(b)$. Then only one agent at each level below and including level q is truly affected by agent i . At level q the agent whose bid is $b_{(n-\sum_{l=q}^{m-1} A_l)}$ is moved from level q to $q-1$. The change in the value this agent receives is $(d_q - d_{q-1})b_{(n-\sum_{l=q}^{m-1} A_l)}$. Likewise, exactly one agent is affected in the same fashion at each level below q until level-1. Summing over all these effects yields

$$\begin{aligned} & (d_q - d_{q-1})b_{(n-\sum_{l=q}^{m-1} A_l)} + \\ & (d_{q-1} - d_{q-2})b_{(n-\sum_{l=q-1}^{m-1} A_l)} + \\ & \dots + \\ & (d_1 - d_0)b_{(n-\sum_{l=1}^{m-1} A_l)} + \\ & (d_0 - d_0)b_{(n-\sum_{l=0}^{m-1} A_l)} \end{aligned}$$

But this is exactly the right hand side of (17), $p_q(b)$, as recursively defined in (8) and (9). ■

The next corollary follows directly from Lemma 1 and the definition in (16).

COROLLARY 1: Agents’ utility function defined in (10) is

equivalent to

$$u_i(b, \theta_i) = \theta_i x_i^*(b) + \sum_{j=1, j \neq i}^n v_j(x^*(b), b_j) - \sum_{j=1, j \neq i}^n v_j(x_{-i}^*(b_{-i}), b_j). \quad (18)$$

Using Corollary 1, we can prove the incentive compatibility result in Proposition 1.

Proof: Let (θ_i, θ_{-i}) denote the truth-telling strategy profile for all agents. For (θ_i, θ_{-i}) to be a (weakly) dominant strategy equilibrium, we must show that for all agent i , $i \in \{1, \dots, n\}$,

$$u_i((\theta_i, b_{-i}), \theta_i) \geq u_i((b_i, b_{-i}), \theta_i) \quad \forall b_i \in \mathcal{B}_i, b_{-i} \in \mathcal{B}_{-i}. \quad (19)$$

By the *revelation principle*, it suffices to show that the truth-telling strategy is a dominant strategy in a *direct revelation mechanism*. A direct revelation mechanism is a game in which the only strategy of each agent is to announce her type. The revelation principle states that a dominant strategy equilibrium of any Bayesian game can be represented by an equilibrium in a direct revelation mechanism. This principle has been enunciated by many researchers, including Gibbard [7], Green and Laffont [9], Dasgupta et al. [3], and Myerson [20]. In $\mathcal{M}_{\text{SPAC}}$, $\mathcal{B}_i = \Theta_i$ for all agent i , $i \in \{1, \dots, n\}$. Therefore, we need to show that for all agent i , $i \in \{1, \dots, n\}$,

$$u_i((\theta_i, \theta_{-i}), \theta_i) > u_i((b_i, \theta_{-i}), \theta_i) \quad \forall b_i \in \mathcal{B}_i, \theta_{-i} \in \Theta_{-i}. \quad (20)$$

Suppose that for some agent i , θ_i is not a dominant strategy. Then there exists $b_i \neq \theta_i$ such that

$$u_i((b_i, \theta_{-i}), \theta_i) > u_i((\theta_i, \theta_{-i}), \theta_i).$$

Substituting from the agent payoff function in (18) in Corollary 1, we have

$$\begin{aligned} & \theta_i x_i^*(b_i, \theta_{-i}) + \sum_{j \neq i} v_j(x^*(b_i, \theta_{-i}), \theta_j) - \sum_{j \neq i} v_j(x_{-i}^*(\theta_{-i}), \theta_j) \\ & > \\ & \theta_i x_i^*(\theta_i, \theta_{-i}) + \sum_{j \neq i} v_j(x^*(\theta_i, \theta_{-i}), \theta_j) - \sum_{j \neq i} v_j(x_{-i}^*(\theta_{-i}), \theta_j) \end{aligned}$$

Substituting from the definition of $v_i(\cdot)$ in (16), we have

$$\sum_{i=1}^n \theta_i x_i^*(b_i, \theta_{-i}) > \sum_{i=1}^n \theta_i x_i^*(\theta_i, \theta_{-i})$$

which contradicts $x^*(\cdot)$ satisfying the maximization condition defined in (5).

Thus, $b_i = \theta_i$. Therefore, $\mathcal{M}_{\text{SPAC}}$ is dominant strategy incentive compatible. ■

APPENDIX B PROOF OF PROPOSITION 2

We prove that $\mathcal{M}_{\text{SPAC}}$ always gives each agent nonnegative utility (payoff).

Proof: Suppose agent i , $i \in \{1, \dots, n\}$, is admitted to QoS level- q , $q \in \{0, \dots, m-1\}$. We can transfer agent i 's utility function, recursively defined in (8) and (9), as follows:

$$\begin{aligned} u_i(b, \theta_i) &= \theta_i d_q - p_q(b) \\ &= \theta_i d_q - (d_q - d_{q-1}) b_{(n - \sum_{l=q}^{m-1} A_l)} \\ &\quad - (d_{q-1} - d_{q-2}) b_{(n - \sum_{l=q-1}^{m-1} A_l)} \\ &\quad - \dots \\ &\quad - (d_1 - d_0) b_{(n - \sum_{l=1}^{m-1} A_l)} \\ &\quad - (d_0 - d_0) b_{(n - \sum_{l=0}^{m-1} A_l)} \end{aligned} \quad (21)$$

Rearranging (21) yields

$$\begin{aligned} u_i(b, \theta_i) &= d_q (\theta_i - b_{(n - \sum_{l=q}^{m-1} A_l)}) \\ &\quad + d_{q-1} (b_{(n - \sum_{l=q}^{m-1} A_l)} - b_{(n - \sum_{l=q-1}^{m-1} A_l)}) \\ &\quad + \dots \\ &\quad + d_1 (b_{(n - \sum_{l=2}^{m-1} A_l)} - b_{(n - \sum_{l=1}^{m-1} A_l)}) \\ &\quad + d_0 (b_{(n - \sum_{l=1}^{m-1} A_l)} - b_{(n - \sum_{l=0}^{m-1} A_l)}) \\ &\quad + d_0 b_{(n - \sum_{l=0}^{m-1} A_l)} \end{aligned} \quad (22)$$

Since $0 \leq d_0 < d_1 < \dots < d_{m-1}$, $b_{(1)}, b_{(2)}, \dots, b_{(n)}$ is the order statistics corresponding to b_1, b_2, \dots, b_n , and $\theta_i = b_i$ by Proposition 1, we get $u_i(b, \theta_i) \geq 0$. ■

ACKNOWLEDGMENT

This work was supported by DARPA Contract No. I30602-01-2-0548.

REFERENCES

- [1] E. H. Clarke, "Multipart Pricing of Public Goods," *Public Choice*, vol. 2, pp. 19–33, 1971.
- [2] R. Cocchi, D. Estrin, S. Shenker, and L. Zhang, "Pricing in Computer Networks: Motivation, Formulation, and Example," *IEEE/ACM Transactions on Networking*, vol. 1, no. 6, pp. 614–27, Dec. 1993.
- [3] P. Dasgupta, P. Hammond, and E. Maskin, "The Implementation of Social Choice Rules: Some General Results on Incentive Compatibility," *Review of Economic Studies*, vol. 46, pp. 185–216, 1979.
- [4] R. Edell and P. Varaiya, "Providing Internet Access: What We Learn from INDEX," *IEEE Network*, vol. 13, no. 5, pp. 18–25, Sep.–Oct. 1999.
- [5] F. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness, and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–52, 1998. [Online]. Available: <http://www.statslab.cam.ac.uk/~frank/rate.html>
- [6] F. Kelly, "Models for a self-managed Internet," *Philosophical Transactions of the Royal Society*, vol. A358, pp. 2335–48, 2000. [Online]. Available: <http://www.statslab.cam.ac.uk/~frank/smi.html>
- [7] A. Gibbard, "Manipulation of Voting Schemes," *Econometrica*, vol. 41, pp. 587–601, 1973.
- [8] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, 1999. [Online]. Available: <http://www.statslab.cam.ac.uk/~frank/evol.html>
- [9] J. R. Green and J.-J. Laffont, "Characterization of Satisfactory Mechanisms for the Revelation of Preferences for Public Goods," *Econometrica*, vol. 45, pp. 427–38, 1977.

- [10] T. Groves, "Incentives in Teams," *Econometrica*, vol. 41, pp. 617–31, 1973.
- [11] G. Hardin, "The Tragedy of the Commons," *Science*, vol. 162, pp. 1243–48, Dec. 1968.
- [12] H. Jiang and S. Jordan, "A Pricing Model for High Speed Networks with Guaranteed Quality of Service," in *Proceedings of INFOCOM 1996*, Mar. 1996, pp. 888–95.
- [13] V. Jacobson, "Congestion Avoidance and Control," in *Proceedings of SIGCOMM'88*. Stanford, CA: ACM, Aug. 1988.
- [14] Y. Liu, W. Gong, and P. Shenoy, "On the Impact of Concurrent Downloads," in *Proceedings of the 2001 Winter Simulation Conference*, Arlington, VA, Dec. 2001, pp. 1300–05.
- [15] S. H. Low, "Equilibrium Allocation and Pricing of Variable Resources Among User-Suppliers," *Performance Evaluation*, vol. 34, no. 4, pp. 207–25, 1998.
- [16] —, "Equilibrium Bandwidth and Buffer Allocations for Elastic Traffic," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 373–83, 2000.
- [17] A. Odlyzko, "Paris Metro Pricing for the Internet," in *Conference on Electronic Commerce (SIGecom)*, 1999, pp. 140–47.
- [18] J. K. MacKie-Mason, L. Murphy, and J. Murphy, "Responsive Pricing in the Internet," in *Internet Economics*, L. W. McKnight and J. P. Bailey, Eds. Cambridge, MA: The MIT Press, 1997, pp. 279–303.
- [19] J. MacKie-Mason and H. Varian, "Pricing the Internet," in *Public Access to the Internet*, B. Kahin and J. Keller, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [20] R. B. Myerson, "Incentive Compatibility and the Bargaining Problem," *Econometrica*, vol. 47, pp. 61–73, 1979.
- [21] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," IETF RFC 2475, Dec. 1998.
- [22] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," IETF RFC 2597, Jun. 1999.
- [23] J. Shu, D. Jaffe, J. Walrand, and P. Varaiya, "Network Service Management via SNT," Electronics Research Laboratory, University of California at Berkeley, Berkeley, CA, Tech. Rep., 2002.
- [24] H. R. Varian, "Estimating the Demand for Bandwidth," University of California at Berkeley, Aug. 1999. [Online]. Available: <http://www.sims.berkeley.edu/~hal/Papers/wtp>
- [25] W. Vickrey, "Counterspeculation, Auctions, and Competitive Sealed Tenders," *Journal of Finance*, vol. 16, no. 1, pp. 8–37, March 1961.