Information Systems Frontiers 5:1, 29–37, 2003

*Mechanism Design for Networking Research**

Jun Shu
Department of Industrial Engineering and Operations Research, University of California at Berkeley, USA
E-mail: jshu@ieor.berkeley.edu

Pravin Varaiya
Department of Electrical Engineering and Computer Science, University of California at Berkeley, USA
E-mail: varaiya@eecs.berkeley.edu

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

Key Words. *pricing, game theory, mechanism design, VCG mechanism, congestion control resource allocation, economical efficiency, QoS, DiffServ, MPLS*

1. 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 (Edell and Varaiya, 1999). An inevitable consequence is the emergence of congestion externality.

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 when there is no congestion. However, when there is congestion, packets may be dropped and delayed, and delivery guarantee become variable. Different applications require different levels of guarantee quality of service (QoS). QoS is measured in terms of delay, jitter, and loss.

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. Bundling becomes the only way to survive. Witness the merger between American-On-Line and Time-Warner—a bundling between network service and content. The jury is still out on whether this is indeed a viable alternative.

The behavior of a network depends on the aggregated traffic load of the network which is the result of many users’ individual decisions on how to use the network. These decisions are affected by the incentives users face. We bring price into network design as signal to control a network.

Our work is inspired by MacKie-Mason and Varian who in 1994 proposed a “smart market” mechanism (MacKie-Mason and Varian, 1994) that suggests an auction based scheme to price congestion. However, the “smart market” mechanism remains a preliminary proposal lacking many crucial aspects of real networks. Our mechanism, 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.

*This work was supported by DARPA contract No. 130602-01-2-0548. On expanded, revised version will be presented at INFOcom 2003.

The rest of the paper is organized as follows. In Section 2, we illustrate the incentive issue for network congestion control. Section 3 presents the SPAC mechanism. We apply SPAC mechanism to a pricing scheme for a network architecture in Section 4. Section 5 concludes the paper with a summary.

2. Incentives for Congestion Control

The Internet provides congestion control through the Transport Control Protocol (TCP) (Van Jacobson, 1988). Congestion is inferred from the absence of acknowledgement for packets received. When there is no congestion, the sender slowly but continuously increases the rate of sending packets. As soon as congestion occurs, the sender drastically decreases the sending rate. The increase and drastic decrement, known as multiplicative decrease and additive increase, upon congestion signal is the key to this algorithm.

If everyone uses TCP (which is still being improved by network engineers), congestion could be managed, however, there is one problem: there is no intrinsic incentive for a user to use the congestion control algorithm. In the next two subsections, we formally illustrate this point from a game theory perspective.

2.1. Two-user case

Let us consider a hypothetical scenario in which there are only two users who are sending packets, at rate 1, through a bottleneck. The bottleneck can allow one packet to pass through in each time interval. When there are two packets competing, the bottleneck randomly lets one packet pass and discards the other one. We model how the two users might interact in a game of strategic form. The game is played on TCP acknowledgement timeout for a packet waiting to be sent in the next time interval.

Each of the two users may choose from one of two opposing strategies. Strategy I is to follow an authentic TCP congestion control scheme which should decide not to send the packet because the timeout has occurred. We will refer to strategy I as “follow-TCP”. Strategy II is to send the packet despite the timeout signal. We will refer to strategy II as “cheat-TCP” since it represents a more aggressive behavior that amounts to cheating in a community adhering to TCP.

We define the payoff function for the two users as the probability of the packet getting through the bottleneck in the next time interval. Obviously, if the packet is not sent at all, the payoff will be zero. If the packet is sent when there is no other packets being sent, the payoff will be one. When there are two packets competing for the bottleneck, we assume that the probability of each packet getting through is fifty-fifty.

Table 1. *Congestion in a two-user game*

		User 2	
		Follow-TCP	Cheat-TCP
User 1	Follow-TCP	0, 0	0, 1
	Cheat-TCP	1, 0	0.5, 0.5

The bi-matrix in Table 1 summarizes the different payoffs to the two users under all possible outcomes of the game. By convention, each cell of the bi-matrix first lists the payoff to the row player, i.e., User 1, and then the payoff to the column player, i.e., User 2. If User 1 chooses follow-TCP strategy and User 2 chooses cheat-TCP strategy, for example, then User 1 receives payoff 0 and User 2 receives payoff 1. When both users choose cheat-TCP strategy, their payoffs are 0.5 and 0.5.

Observe that User 1 is always better-off by choosing the cheat-TCP strategy. So is User 2 by choosing the same strategy. Therefore, the strategy profile in which both users choose to cheat-TCP is a Nash equilibrium of this game. Furthermore, in any other combinations of strategies, a user has an incentive to deviate from that combination. It is not a stable state for both users to choose the follow-TCP strategy.

Although our two-user game model may be accused of inconsistency with reality in a variety of ways, it should be clear that the model formally demonstrates the lack of incentives in the TCP type of congestion control schemes.

2.2. Many-user case

Now consider a packet-switched network with n users. Let x_i denote the rate of sending packets by user i, $i \in \{1, \ldots, n\}$. Let L denotes the total rate, $L = \sum_{i=1}^{n} x_i$. Hence, L is a measure of the network load. Assume that (a) the price of sending packets at unit rate is a constant per unit rate, p; (b) the value of sending packets at unit rate is a function of the network load, $v(L)$; (c) there is a maximum rate the network can accommodate, $L_{\max}$; (d) $v(L) > 0$ for $L < L_{\max}$ and $v(L) = 0$ for $L \geq L_{\max}$; (e) $v'(L) < 0$ and $v''(L) < 0$ for $L < L_{\max}$; and (f) rates are continuously divisible.

Our objective is to reach an optimal level of network load. A central planner of network usage, acting on behalf of all the users, would solve

$$\max_{0 \le L \le \infty} Lv(L) - Lp,$$

the first-order condition for which is

$$v(\hat{L}) + \hat{L}v'(\hat{L}) - p = 0 \qquad (1)$$

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

However, in reality, such a central planner does not exist 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 phenomenon with a strategic game.

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

$$x_i v(L) - px_i \qquad (2)$$

Let $(x_1^*, \ldots, x_n^*)$ be a Nash equilibrium. Then, for each user i, x_i^* must maximize the payoff function (2) given that the other users choose $(x_1^*, \ldots, x_{i-1}^*, x_{i+1}^*, \ldots, x_n^*)$, sum denoted by x_{-i}^*. The first-order condition for maximizing the payoff function (2) is

$$v(x_i + x_{-i}^*) + x_i v'(x_i + x_{-i}^*) - p = 0 \qquad (3)$$

Summing over all n users' first-order conditions yields

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

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

It is straightforward to prove that $L^* > \hat{L}$, which says that 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 his or her own incentives, not the effect of his or her actions on the other users.

2.3. Information asymmetry

Interestingly, despite the incentive issue, 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. Consequently, whether one uses Microsoft Internet Explorer or Netscape Navigator, whether one is on Windows or Unix, TCP is working transparently to network users. Few people are aware of this incentive issue. Even fewer people know how to change their computer 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; and today's good fortune depends upon a few "benevolent" dictators (i.e., software companies like Microsoft). The knowledge will spread; and once known, if there is a will there will be a way.

In fact, a number of applications have appeared to explore the incentive issue in the TCP congestion control scheme. They include FlashGet, GolZilla, ReGet, Download Accelerator, GetRight, GetSmart, and Download Devil (Liu, Gong, and Shenoy, 2001). 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. Although each connection is still using TCP, the applications behave in a much more aggressive way that in fact abandons TCP congestion control without changing it.

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

3. Smart Pay Admission Control Mechanism

Since congestion is caused by too many users competing for 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 by the statistical guarantee the packets receive. This means that the higher value a user put on the service, the better statistical guarantee of delivery the user will get. The rest of this section discuss our mechanism that achieves this goal.

3.1. Mechanism definition

Formally, a *mechanism* is a game which includes players, players' strategies, and players' payoff functions. We define the Smart Pay Admission Control (SPAC) mechanism, denoted by $\mathcal{M}_{\text{SPAC}}$, as follows.

3.1.1. Players. There are $n + 1$ players including n agents, denoted by player $i = 1, \ldots, n$, and, one principal, denoted by player $i = 0$.

3.1.2. 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 delivering 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 $k = 0, 1, \ldots, m-1$. Let d_k denote the service delivery rate of kth level, and $d = (d_0, d_1, \ldots, d_{m-1})$ where $d_0 < d_1 < \cdots < d_{m-1}$.

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

3.1.3. Agents' actions. An agent receives the service at QoS level $k = 0, 1, \ldots, m-1$. 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 his or her value of the service. The *announced values*, also referred to as bids, are denoted by $b = (b_1, \ldots, b_i, \ldots, b_n)$ where b_i is the bid for agent i. The bids are disclosed to at least the principal. The *true values*, also referred to as types, which are each agent's private information, are denoted by $\theta = (\theta_1, \ldots, \theta_i, \ldots, \theta_n)$ where θ_i is the true value (type) of agent i. Let B_i denote the space of allowable bids and Θ_i the space of agent type for agent i.

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

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

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

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

where,

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

The solution $x^*(b)$, a function of b, is efficient and feasible if

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

subject to the capacity constraint

$$\sum_{i=1}^{n} Q_k(x_i) \le A_k \quad \forall k = 1, \ldots, m-1 \tag{6}$$

and subject to the universal service coverage constraint

$$\sum_{k=0}^{m-1} A_k \ge n. \tag{7}$$

3.1.5. Congestion price. Based on agents' bids, the principal computes a congestion price for each level of QoS. Let $p = (p_0, p_1, \ldots, p_{m-1})$ denote all prices, where

$$p_0(b) \equiv 0 \tag{8}$$

and

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

3.1.6. Agents' payoff. Agent i's payoff function is

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

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.

3.2. 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 amount of agents the network can serve at each level of QoS, that is A_k for $k = 0, 1, \ldots, m-1$, 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 Eqs. (5)–(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 A_{m-2} bidders to level d_{m-2}, etc. until all the bidders are admitted.

Calculation of the congestion prices as defined in Eqs. (8) and (9) is straightforward after sorting all the bides. Notice that according to the admission control algorithm, $b_{(n-\sum_{l=m-1}^{k} A_l)}$ is the highest bid whose agents are not admitted to kth or above levels of QoS.

3.3. Agents' strategies

A strategy for agent i is his or her bid. The strategy space is $[0, \infty)$. A rational agent chooses a strategy that would maximize the agent's payoff which is dependent on the actions taken by all agents . We claim that given the special form of payoff function defined as in Eq. (10), the optimal strategy for every agent, regardless of other agents' strategies, is to announce his or her true value. That is, $b_i = \theta_i$ is the solution to maximize Eq. (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* if it is a dominant strategy for every agent to announce his or her true value (or type). We now prove that $\mathcal{M}_{\text{SPAC}}$ is dominant strategy incentive compatible.

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

$$
\begin{aligned}
x^*_{-i}(b_{-i}) = \arg \max_{x_j \in \{d_0, d_1, \dots, d_{m-1}\}} & \sum_{j=1, j\neq i}^{n-1} b_j x_j \\
\text{subject to} \quad & \sum_{j=1, j\neq i}^{n-1} Q_k(x_j) \leq A_k \quad \forall k = 1, \dots, m-1 \\
\text{and} \quad & \sum_{k=0}^{m-1} A_k \geq n-1. \qquad (11)
\end{aligned}
$$

Let $v_j(x, b_j)$ denote the (declared) value to agent j from the service received. That is,

$$v_j(x, b_j) = b_j x_j \quad \forall j \in \{1, \dots, n\}. \qquad (12)$$

Lemma 3.1. *For all* $i \in \{1, \dots, n\}$ *there exists* $q \in \{0, \dots, m-1\}$ *such that* $Q_q(x_i^*(b)) = 1$, *and*

$$\sum_{j\neq i} v_j(x^*_{-i}(b_{-i}), b_j) - \sum_{j \neq i} v_j(x^*(b), b_j) = p_q(b). \qquad (13)$$

Proof: The left hand side of Eq. (13) is the impact that 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, and, W.L.G., 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 effected by agent i. At level q the agent whose bid is $b_{(n-\sum_{l=m-1}^{q} A_l)}$ is moved from level q to $q-1$. The change in the value this agent receives is $(d_q - d_{q-1}) b_{(n-\sum_{l=m-1}^{k} A_l)}$. Likewise, exact one agent is effected in the same fashion at each level below q until level-1. Summing over all these effects yields

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

But this is exactly $p_q(b)$ as recursively defined in Eqs. (8) and (9). □

The next corollary follows directly from Lemma 3.1 and the definition in Eq. (12).

Corollary 3.1. *Agents' payoff function defined in Eq.* (10) *is equivalent to*

$$
\begin{aligned}
u_i(b, \theta_i) = \theta_i x_i^*(b) &+ \sum_{j \neq i} v_j(x^*(b), b_j) \\
&- \sum_{j \neq i} v_j(x^*_{-i}(b_{-i}), b_j). \qquad (14)
\end{aligned}
$$

Using Corollary 3.1, we can prove the incentive compatibility result.

Theorem 3.1. *The Smart Pay Admission Control mechanism*, $\mathcal{M}_{SPAC}$, *is dominant strategy incentive compatible.*

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, $i \in \{1, \dots, n\}$,

$$
\begin{aligned}
u_i((\theta_i, b_{-i}), \theta_i) &\geq u_i((b_i, b_{-i}), \theta_i) \\
&\forall b_i \in B_i, b_{-i} \in B_{-i}. \qquad (15)
\end{aligned}
$$

By the *revelation principle*, it suffices to show that the truth-telling strategy is a dominant strategy in a *direct revelation mechanism*.[1] That is,

$$
\begin{aligned}
u_i((\theta_i, \theta_{-i}), \theta_i) &> u_i((b_i, \theta_{-i}), \theta_i) \\
&\forall b_i \in B_i, \theta_{-i} \in \Theta_{-i}. \qquad (16)
\end{aligned}
$$

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 Eq. (14) in Corollary 3.1, we have

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

Substituting from the definition of $v_i(\cdot)$ in Eq. (12), 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 Eq. (5).

Thus, $b_i = \theta_i$. Therefore, $\mathcal{M}_{\text{SPAC}}$ is strategy-proof. □

In fact, $\mathcal{M}_{\text{SPAC}}$ is a case of the Clarke-Grove mechanism and can be viewed as a generalized Vickrey Auction (1961).

3.4. *Agents' participation*

Suppose that agents' participation in the SPAC mechanism is voluntary. To participate in a mechanism, an agent must agree to be bound by the rules in the mechanism. An agent would participate in a mechanism only if his or her expected payoff from the participation is at least as good as that from not participating in the mechanism. This condition is known as the *participation constraint*.

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

- *ex ante* participation constraints

$$E_{b_i}\left\{E_{b_{-i}}[u_i(b, \theta_i) \mid b_i]\right\} \geq E_{\theta_i}[\bar{u}_i(\theta_i)] \quad \forall i \in \{1, \ldots, n\} \quad (17)$$

- *interim* participation constraints

$$E_{b_{-i}}[u_i(b, \theta_i) \mid b_i] \geq \bar{u}_i(\theta_i) \quad \forall i \in \{1, \ldots, n\} \quad (18)$$

- *ex post* participation constraints

$$u_i((\theta_i, \theta_{-i}), \theta_i) \geq \bar{u}_i(\theta_i) \quad \forall i \in \{1, \ldots, n\} \quad (19)$$

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.

Suppose that

$$\bar{u}_i(\theta_i) = \bar{d}\theta_i \quad \forall i \in \{1, \ldots, n\} \quad (20)$$

where $\bar{d}$ is the service delivery rate agent i can receive without participating in the SPAC mechanism. 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 $d = (d_0, d_1, \ldots, d_{m-1})$, and
- The distributions of agent types, θ_i for all $i \in \{1, \ldots, n\}$.

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.

Let $q, q \in \{0, \ldots, m-1\}$, be a random variable that denote the QoS level assigned to agent i. According to the definition in Eq. (10), agent i's payoff function from the mechanism is

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

If the distributions of agents' types are unknown, then we can say little about q. However, the principal can satisfy the *ex post* participation constraints 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.

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

3.5. *Principal's rationality*

Why would a service provider adopt the SPAC mechanism in the first place? The rationality for the principal, i.e., the service provider, is similar to that of the agents' participation constraints. Compared to the current practice of 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 congestion fee, and provide better service satisfaction to users.

4. SPAC Based Pricing for DiffServ

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

In DiffServ architecture (Blake et al., 1998), traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different aggregates. Each aggregate is identified by a specific DiffServ codepoint in 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 the service differentiation achieved. Scalability is achieved through three major features of the DiffServ model. Firstly, service differentiation is given to packet aggregates rather than individual microflows. Secondly, 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 the states of the traffic.

Our DiffServ pricing scheme consists of two parts: pricing for traffic profiles and pricing for out-of-profile traffic streams.

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 the 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. On the other hand, 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 even 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 is selling 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 the users and service providers. The type and number of marketplaces for TPs will develop as the network service business evolve and mature.

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 the SPAC mechanism, $\mathcal{M}_{\text{SPAC}}$, defined in Section 3. 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.

- The service provider decides how many different levels of QoS the network will provide. This is the "service delivery rate" vector, $d = (d_0, d_1, \ldots, d_{m-1})$, defined in $\mathcal{M}_{\text{SPAC}}$.
- The different service delivery rates are statistically guaranteed through a Resource Manager which 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.
- An Automatic Agent (AA), 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.
- A Bandwidth Broker (BB), based on all of users' bids, calculates the congestion fee for each QoS level, i.e., the same $p = (p_0, p_1, \ldots, p_{m-1})$ as defined in

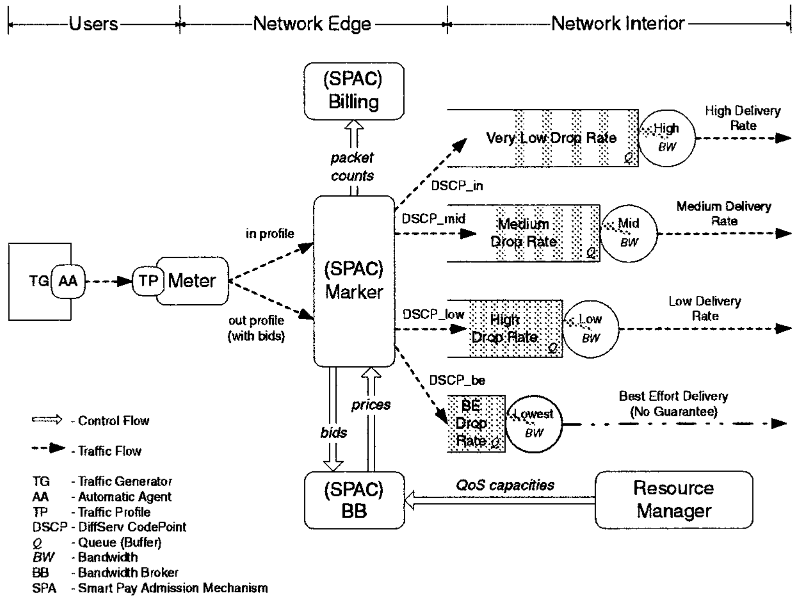

Fig. 1. *SPAC pricing for DiffServ.*

$\mathcal{M}_{\text{SPAC}}$. BB receives from the Resource Manager current capacity information for different levels of QoS, i.e., the $\{A_k, k = 0, \ldots, m-1\}$.

- 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 $\mathcal{M}_{\text{SPAC}}$.
- A Billing system records the out-profile packet counts from the Marker, and the temporal congestion fee.

Fig. 1 shows a schematic representation of the $\mathcal{M}_{\text{SPAC}}$-based DiffServ pricing. The figure shows one traffic profile and four different levels of QoS. To ensure users' participation in this scheme, as we have discussed in Section 3.4, the lowest quality is that of the best-effort, which is the quality of current Internet. $\mathcal{M}_{\text{SPAC}}$ also ensures that all bids reflect the true value of service perceived by users, and that congestion is well under controlled at the QoS level-1 and above because the right 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 does not even need to be aware of the SPAC mechanism.

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

5. 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 the incentive issue. As the major theme of the paper, we proposed the Smart Pay Admission Control mechanism. We proved the incentive compatibility and participation constraints of this mechanism. Finally, we proposed a pricing structure 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. 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 of networks in which traffic profiles contains routing information and thus congestion is reflected along a path instead of a domain.

With our pricing scheme, congestion control is truly achieved—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 and autonomous 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 expand customer bases.

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 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 sorely 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 Smart Network Toolkit, acronym SNT. A technical paper on SNT is available to interested readers (Shu, 2001).

Note

1. A direct revelation mechanism is a game in which the only strategy of each agent is to announce his/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 (1973), Green and Laffont (1977), Dasgupta et al. (1979), and Myerson (1979).

References

Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W. An architecture for differentiated services. Technical Report RFC 2475, IETF, 1998.

Cocchi R, Shenker S, Estrin D, Zhang L. Pricing in computer networks: Motivation, formulation, and example. *IEEE/ACM Transactions on Networking* 1993;1(6):614–627.

Edell R, Varaiya P. Providing internet access: What we learn from index. *IEEE Network* 1999;13(5):18–25.

Heinanen J, Baker F, Weiss W, Wroclawski J. Assured forwarding PHB group. Technical Report RFC 2597, IETF, 1999.

Jacobson V. Congestion avoidance and control. In: *Proceedings of SIGCOMM'88*, Stanford, CA, 1988, ACM.

Liu Y, Gong W, Shenoy P. *On the Impact of Concurrent Downloads*. Department of Electrical and Computer Engineering, University of Massachusetts, 2001.

MacKie-Mason JK, Murphy L, Murphy J. In: McKnight LW, Bailey JP, eds. *Internet Economics*, The MIT Press: Cambridge, MA, 1997:279–303.

MacKie-Mason JK, Varian HR. *Pricing the Internet*. University of Michigan, 1994.

Oren SS. *Transmission Pricing and Congestion Management: Efficiency, Simplicity and Open Access*. University of California at Berkeley, 2000.

Shu J. NMS Project Annual Report: Smart Network's Toolkit, September 2001.

Vickrey W. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 1961;16(1):8–37.