Predictive Dialing for Outbound Telephone Call Centers

Douglas A. Samuelson

InfoLogix, Inc.
8711 Chippendale Court
Annandale, Virginia 22003

In the late 1980s, I used queuing and simulation to invent predictive dialing, a method to determine when computer-directed outbound telephone dialing systems should dial. I included a real-time estimation updating feature that was highly robust against sudden changes in the system's operating environment; thorough validation to ensure that the models tracked all important features of the real systems; and a modular software design that allowed "plug-in" replacement of the control software, eliminating debugging of field upgrades. The improved systems kept operators busier and drastically reduced the number of calls the systems abandoned because no operator was available to talk to the answering party. This invention was critical to the success, in the late 1980s, of International Telesystems Corporation (ITC), a small company founded in 1984, which a competitor, EIS International, bought in 1993 for approximately $12 million.

Computer-based outbound telephone-dialing systems are widely used in telemarketing, debt collection, and organizational fund raising. These systems dial automatically and connect those who answer to live sales or collection representatives, while logging other results. Sometimes someone answers when no representative is available. Typically the system immediately abandons (hangs up

0092-2102/99/2905/0066/$05.00
1526–551X electronic ISSN
This paper was refereed.

QUEUES—SIMULATION
COMPUTERS/COMPUTER SCIENCE—SOFTWARE

INTERFACES **29:** 5 September–October 1999 (pp. 66–81)

on) such a call, with the caller paying for telephone charges and wasting representatives' time and—more important—with the answering party suffering a nuisance.

In 1986–1987, as a senior analyst and manager of International Telesystems Corporation (ITC), a manufacturer of such systems, I invented what is now known as predictive dialing: an improved method of deciding when to dial, to keep representatives busy and to reduce the number of abandoned calls. The key idea is to anticipate when representatives will complete calls and to time new calls so that the next called person will answer (arrive) shortly after a representative becomes idle. This was a major departure from the usual practice in the industry at the time, which was to begin a new dialing attempt every n seconds (usually three to 10). Those controlling such systems could vary the interval between dialing-attempt starts, depending on how busy the representatives were; typically, they made these adjustments too slowly, so the systems oscillated between too much idle time and too many abandoned calls.

If all we wanted was to keep the representatives as busy as possible, we could simply dial every available line all the time. This, however, would result in large numbers of abandoned calls. Most call-center managers want to keep abandoned calls under five percent of completed calls; some insist that they want no abandoned calls at all. We could ensure no abandoned calls by dialing one line per idle representative, and only for idle representatives, but this typically results in keeping representatives busy less than 40 minutes per hour. Most call-center managers insist on 50 minutes per hour and would prefer more than 55. These conflicting objectives make this problem challenging.

The proportion of call attempts answered quadruples from 5:00 p.m. to 6:00 p.m.

What I proposed was to find better pacing (scheduling of dialing attempts) by collecting and analyzing, in real time, data on the proportion of call attempts that are answered, durations of time from call initiation to answer, and durations of service. Queuing theory provides some guidance in choosing policies, but the method must also be robust in handling sudden changes in the situation. For example, when we call residences on weekdays, the proportion of call attempts answered typically quadruples from 5.00 p.m. to 6.00 p.m. Durations of completed calls also change over time for many applications: for example, if the calls are directed at a specific member of the household who is more likely to be home at 6.00 p.m. than at 5.00 p.m.

The method must also adjust nearly instantaneously to representatives logging on and off, terminals failing, long-distance carriers' switching equipment having problems, and various other surprises. There is no steady state, and there is little useful guidance available from previous similar experience, even in a given system on consecutive days. It is quite easy to design a control method that will cause a calling system to oscillate between over- and underutilization of representatives; the control method must adjust smoothly to various changes in the situation.

Implementation also must be robust against coding errors. Downtime in the field is expensive and quickly undermines customers' confidence in the company. Ideally, therefore, the code controlling the pacing of dialing attempts should be a plug-in module in the system's operating software and should provide acceptable default behavior in case of data problems or other real-time faults.

Problem Statement and Preliminary Analysis

As I formulated it, the problem was to determine when to commence dialing attempts to maximize the number of calling attempts per hour (or, equivalently, the utilization rate of representatives) subject to an upper bound on the proportion of calls that end up being abandoned because no representative is available when the party answers.

A conversation with the right party lasts one to three minutes.

The solution idea is to anticipate service completions and to synchronize calling attempts with them. Ideally, we want to begin a calling attempt so that, if the party answers, he or she will answer just after a representative, now busy, finishes his or her current conversation (service). This means we must estimate the time remaining until service completion and determine the amount of time by which to anticipate service completion. For example, if we know that service will end 20 seconds from now and that the party will answer 15 seconds from when we start dialing, we start dialing five seconds from now.

Since the actual duration of service and the time from start of dialing to connect are random, what we do is trade off the risk of obtaining the new party too early, resulting in an abandoned call and the need to start dialing again, and the cost of waiting too long, resulting in unnecessary idle time for the representatives. The decision variable here is the amount of time by which to anticipate service completions.

In addition, given that dialing attempts do not necessarily result in answers, we may want to dial more than one party at once. If two or more answer, we will have one or more abandoned calls; if none answers, we will have idle representative time. So the number of calls to attempt at once is another decision variable.

Review of Prior Studies

The literature I surveyed was not much help to me. Most prior researchers assumed exponential service times, which means that knowing how long a service has lasted provides no information about how much longer it will last. Most prior work also was based on arrivals according to processes the system could not influence: The decision was to admit or reject an arriving customer, not whether to try to acquire one. Accordingly, the optimal policies took the form of *control limits*: For some calculated number n, admit the new customer if there are n or fewer customers in the system now, and reject otherwise.

The most promising approach in the literature was to apply dynamic programming to semi-Markov-decision-process models to generate optimal policies. (Stidham [1986] provides the best summary of SMDP work in the early to mid-1980s; Puterman [1994] gives an excellent

and more recent summary of SMDP in general.) While the examples available used exponential service times and therefore generated control-limit policies, this method offered the possibility of handling other service-time distributions without becoming completely intractable.

As it turned out, however, applying SMDP was not practical in this case. I consulted Kyung Jo, an expert with SMDP models. (He was a faculty member at George Mason University at the time, having recently completed his doctorate under Professor Stidham's direction.) I soon learned that the coding problems were ferocious, as the method requires specifying a reward or cost for each step in the process (completed call, logged no-answer, and so on) and the limited number of model runs he did each took about 40 minutes of CPU time on a Cyber 205—much too compute-intensive for a real-time application!

Modeling Approach

After some experimentation, I found it best to model the process as a closed queuing network (Figure 1). The system's telephone lines are the "customers" of the queuing network, and the states the telephone lines can occupy—dialing, in service, postservice processing, processing nonanswer, waiting to dial—are the nodes of the network. Customers are held at control point P; allowing a customer to proceed from this point begins an acquisition. If the acquisition is unsuccessful (in the case of the call-origination system, this means any result other than an answer), the customer proceeds back around the loop to P. Node D represents, for the model, the delay the customer experiences after an unsuccessful acquisition; this feature allows us to model unsuccessful acquisitions with different durations from successful acquisitions. The successful acquisitions proceed to service node Q, which consists of s servers (operators) plus a waiting area. Q is the node of particular interest. After completing service, the customer returns to control point P via a process that I represent as delay node R, which is simply the processing the system must perform to make the line available for dialing again.

Operators' talk times would increase from 48 minutes per hour to 57.

In front of service node Q is a waiting area with finite capacity; if a customer arrives and finds this waiting area full, it immediately balks and returns to control point P. This event is a turnaway, or abandoned call, or lost party (since the called party, but not the telephone line, leaves the system). We denote as delay node N the customer's transit back to P. This transit consists of the processing the system must perform before dialing the line again.

We may view success or failure of acquisition as assigning customers to one of two classes. One class proceeds through delay node A and then to service node Q; the other customers take alternate route A′ and are processed through (alternate) delay node D. We assume that a given customer, once released from control point P, has probability p of acquiring a party and hence requiring service at the node of interest, Q, and probability $q = 1\text{-}p$ of tra-

versing alternate route A′ and being "served" at node D. We assume further that customers divide into these routings, or classes, randomly and independently, and that a customer's class (or route) is not known before it is released from P. Finally, we treat all service nodes except Q as having ample capacity.

This way of viewing the system, while perhaps not intuitive at first, made tractable mathematical modeling possible. It is required because the number of answering parties is, in effect, infinite, but the number of lines is limited; and we want the model to take into account the constraint imposed by the limited number of lines. The telephone lines are like buses, and answering parties are like passengers whose presence affects where the buses go. Figure 2 portrays the system in these terms. I found that this depiction was more easily understood than Figure 1 by people who had no background in queuing models and other technical aspects; I found this figure especially helpful, therefore, in explaining my method to prospective buyers of ITC's systems.

This model represents a new class of queuing problems, in which customer arrivals do not occur strictly at random but instead occur as responses, which may be random to some extent, to acquisition attempts initiated by the system. We see many potential applications beyond outbound telephone dialing and an interesting set of theoretical questions, but these topics are beyond the scope of this paper. I outline some of the theoretical aspects, however, in the appendix.

Solution Method

ITC's system was able to collect, virtually instantaneously, durations of service, proportions of dialing attempts that resulted in an answer, and durations of suc-

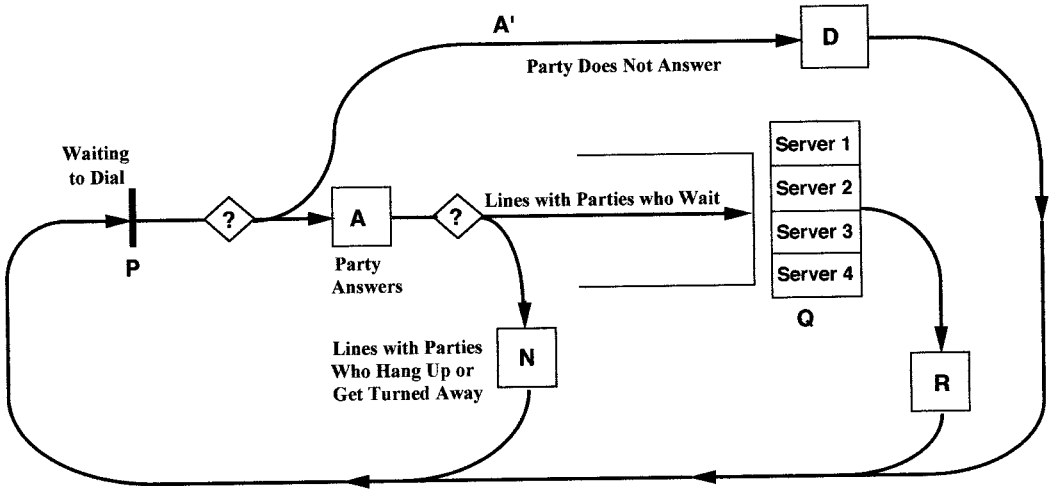

Figure 1: We depict the outbound telephone dialing system as a closed queuing network. The system's telephone lines are the customers; responses other than answering party are treated as selection of the route A′ through delay node D, while lines with answering parties (A) are directed to the servers (telephone representatives) at node Q. If all servers are busy and parties refuse to wait (or the system does not provide for waiting), customers are turned away.

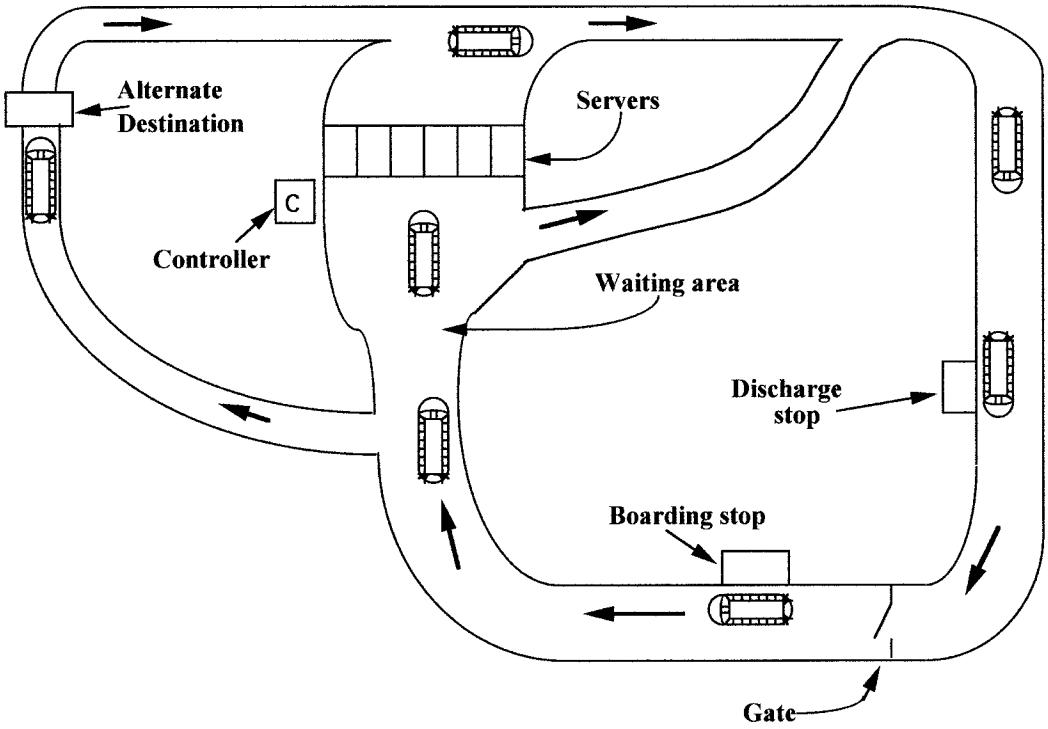

Figure 2: This is another depiction of the system, more intuitive for nontechnical people. The system's telephone lines are carriers (the buses) that pick up passengers (dialed parties). These passengers then decide their destination: the service area (if they answer) or some alternate destination (busy, no answer, telephone company message) and the system responds accordingly. Control consists of deciding when to release the next bus (dial), depending on how congested the service area is.

cessful dialing attempts. Statistical calculations, such as means, maxima, variances, and various quantiles, were also quick and straightforward.

We could then use these data to generate decisions:

—For each server (representative) now serving a called party, compare how long that service has lasted so far against the estimated duration of service to obtain an estimate of time remaining.

—If the estimated time remaining in service is less than or equal to the estimated time to obtain a new called party, count this server as available. (Idle servers are also available.)

—If the number of attempts already in progress is not expected to produce as many additional answering parties as the number of servers expected to be available, begin additional dialing attempts.

I give a format statement of this solution method in the appendix.

Evolution of Solution Details

By extensive trial and error using simulation, I determined that varying the number of attempts to begin at once generated a more volatile response than varying the lead time by which to anticipate service completions. If, for example, 20 percent of

dialing attempts result in completions, it is not good to begin five attempts at once. This is a simple binomial problem: The probability of two or more simultaneous answers is a little more than one-fourth, and the expected number of abandoned calls per set of attempts is a little higher, about 0.33, which would mean about 25 percent of completed calls would be abandoned—an unacceptable result. In general, I found it best to dial two lines at once if the proportion answering was under 33 percent, and three lines if the proportion was under 20 percent—a more conservative approach.

I also learned quickly that mean duration of service and mean time to complete an attempt were poor measures to use: Maximum duration of service—or something close to that—and minimum time to complete an attempt were more meaningful. If we wanted a little more speed and were willing to have a slightly higher percentage of abandoned calls, we simply increased the anticipation time.

For simplicity's sake, I assumed that servers would have identical probability distributions of duration of service. I made the method more realistic later, as additional sales and installations of ITC's systems gave us the opportunity to collect more and better data.

Many applications, especially in debt collections, produce a bimodal distribution of service time: A conversation with the right party lasts one to three minutes, while a conversation with someone else lasts 30 to 90 seconds. In these cases, the mean duration of service is a particularly poor estimate, as it generally falls near the low point between the two modes. I found it better to use the maximum duration observed as the estimate for services that had lasted longer than most short services, and a mixture of the maximum short and the maximum long service, weighted by the relative frequencies, for services that had not yet lasted that long.

It did occur to me that staggering starts of dialing attempts might offer a useful improvement, but this was more complicated both to analyze and to implement, so I postponed considering it. In fact, ITC did adopt this improvement a couple of years later, along with a method—rather difficult with the switching equipment and control program we had in 1987, but easier later—to abandon some dialing attempts in progress when patrons answered. Preferably these would be attempts that had not yet produced the first ring on the called party's telephone, and in 1987, we could not reliably detect the first ring.

Another key element, as we soon discovered, was updating estimates as new experience occurred. We updated estimates every 10 minutes, as this was about the amount of time it took to get 100 or so attempts on a small system. More frequent updates produced more instability, and less frequent updates produced too-slow adjustments; 10-minute updating appeared to be about the best compromise.

I found it best to make immediate adjustments, slowing the system down, when we saw a new longest duration of service or when the proportion answering increased dramatically, but to update more slowly when duration of service appeared to be decreasing or proportion answering seemed to be dropping, indicating

a need to speed up. For these updates, I settled on exponential smoothing, with a parameter of about 0.33, meaning that a new condition had to persist for about half an hour (three updates) to move the estimate most of the way to the new value.

Performance Implications

Crude as some aspects of this solution method were, it promised a dramatic improvement over previous methods. Services typically last 30 to 120 seconds; successful long-distance dialing attempts typically take from 20 to 30 seconds from start to answer. If almost all services (say 98 percent) end within 105 seconds, and we start dialing at 85 seconds, we can save 20 seconds between services (that is, reduce the average time between calls from 25 seconds to five), relative to the wait we would expect if we did not dial until service ended, at a cost of about two to three percent abandoned calls. This means that, for this oversimplified example, operators' talk times would increase from about 48 minutes per hour to 57 minutes per hour—that is, from 80 percent utilization to 95 percent utilization.

For actual situations with more variability, in which eliminating abandoned calls generally meant limiting operators' talk times to around 40 minutes per hour (66 percent utilization), we hoped we could obtain talk times in the 50 to 55 minutes-per-hour range, or 83 to 90 percent utilization.

In simple one-server systems with one acquisition agent (dialer) in which there is no alternate routing (arriving customers either enter service or leave), all acquisition attempts succeed and no waiting is permitted (Figure 3, first part). To maximize throughput and server utilization, subject to turning no customers away, with a control-limit policy, we begin an acquisition whenever a service ends. There is, therefore, always exactly one customer in the acquisition-service subsystem.

In contrast, by anticipating completions of service, we can begin an acquisition while service is still in progress (Figure 3). This policy—if service times are bounded so that the service is certain to end before the acquisition succeeds—preserves the no-turnaways performance while shortening the idle times between services.

EIS's legal counsel estimated the value of the patent as between $1 and $2 million.

Suppose now that we are willing to have some customers turned away to increase throughput. Our only option among control-limit policies would be to increase the control limit from one to two. Then we would always have exactly two customers in the acquisition-service subsystem: either one in service and one in acquisition, or two in acquisition. Each acquisition completed during a service would result in a turnaway, and another acquisition would start immediately. A service completion would also cause another acquisition to begin, even though the other acquisition, already in progress at the service completion, would soon produce a customer (Figure 4).

One possible anticipative policy to increase throughput is to start two acquisition attempts simultaneously when we anticipate a service completion. The earlier of the two attempts to finish would enter ser-

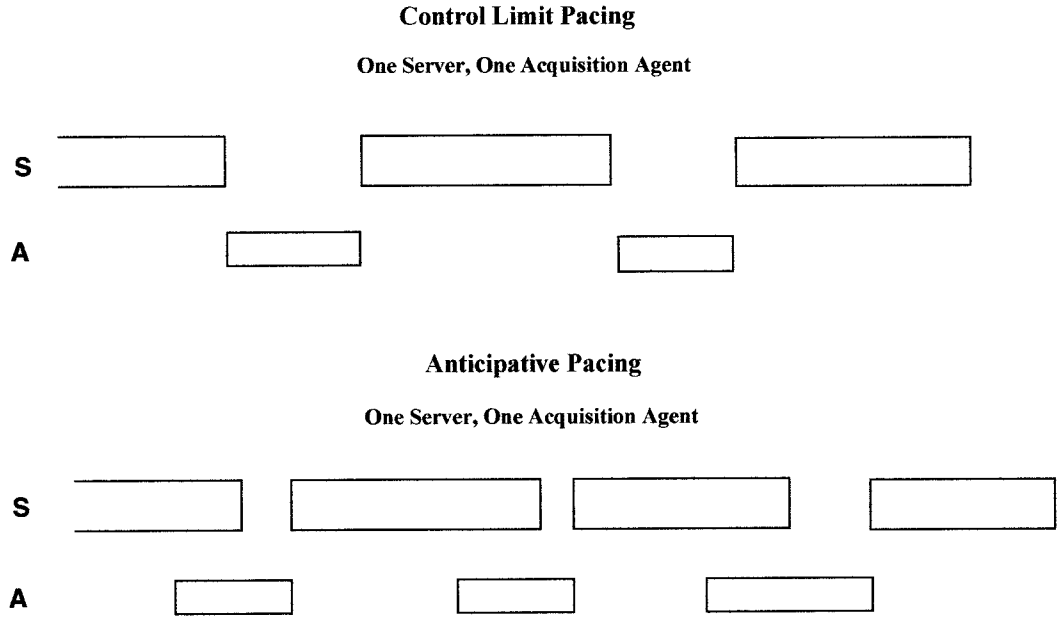

Figure 3: We depict schematically how anticipating service completions improves throughput and server utilization. If we allow one customer in the system at a time, we start an acquisition attempt (A) whenever a service (S) completes. Anticipation reduces the servers' idle time between services.

vice, while the other would be turned away. This is clearly a naive and unsatisfactory policy, but it would still be better than the naive increase of the control limit from one to two in this case (Figure 4). In this case, the anticipative policy produces higher throughput with fewer turnaways than the control-limit policy.

I knew, however, that real life would present unexpected complications. Even the model had shown one surprising result: In some cases, increasing the lead time (that is, the time by which we anticipate a service completion) decreases the servers' utilization and throughput (Figure 5). The first increment of lead time increases throughput, as expected, as the probability of obtaining a new party before the end of service is small, and we save some time by not waiting until the last possible time of service completion. As we continue to increase the lead time, however, the probability of obtaining the next party too soon, forcing us to abandon that party and start the acquisition process over, outweighs the time we expect to save if the service ends early. I was able to prove the occurrence of this phenomenon

for a couple of simple if somewhat unlikely analytical cases, using Markov renewal theory (appendix).

Implementation

To ensure that I had captured all relevant behaviors of the system, I decided that simulation modeling was not sufficient: I wanted as realistic a test bed as possible. Therefore, I worked with our software development group to construct an emulator system including everything except the telephone switching equipment, with the actual control software configured so that the pacing routines were plug-in modules. A subroutine that randomly generated call-attempt results and service times replaced the switching equipment.

For other reasons, this test bed turned out to be perhaps the most important thing we did. It ensured that we hadn't missed any vital features of the actual control software and gave us the opportunity to observe performance problems as they would look in real life, but the biggest benefit appeared only when we installed the program in the field: Since the pacing routines were already written and fully tested as plug-in modules with the control software, there were absolutely no installation problems in the field.

This is not to say that there were no surprises. We did encounter unexpected conditions. One particularly unpleasant one was when a terminal froze, and the system kept obtaining new parties intended for

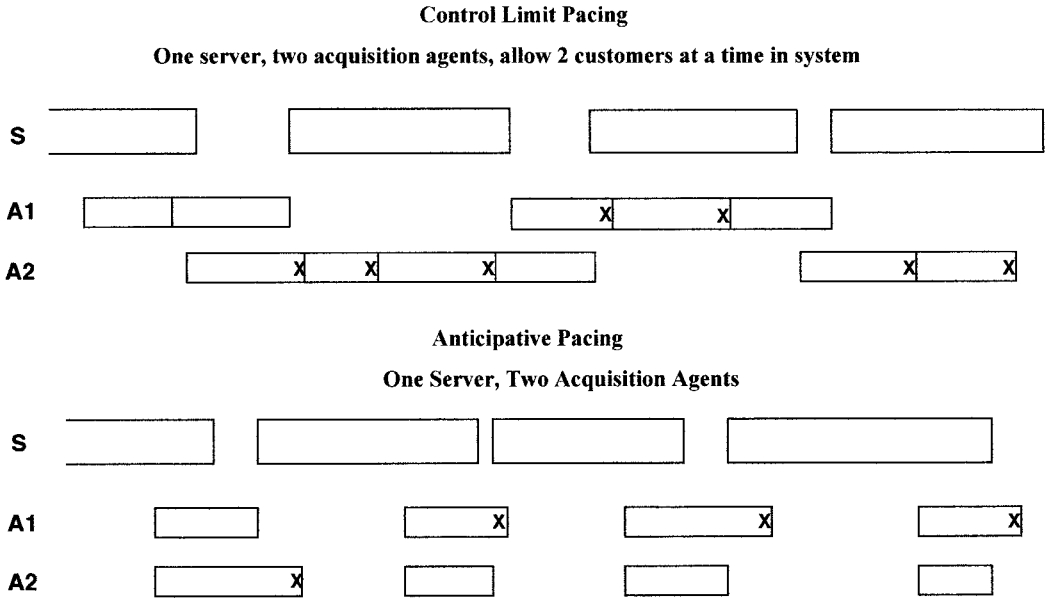

Figure 4: We depict schematically the comparison of control limit versus anticipative policies, this time with two customers allowed in the system at once. An "X" means the acquired customer is turned away. The control-limit method will have one acquisition attempt active at all times and two acquisition attempts when the server is idle, so many answering parties will be turned away. If we start two acquisition attempts every time we anticipate a service completion, one enters service and the other is turned away—a poor policy, but better than its control-limit counterpart, with both fewer turnaways and less idle time between services.

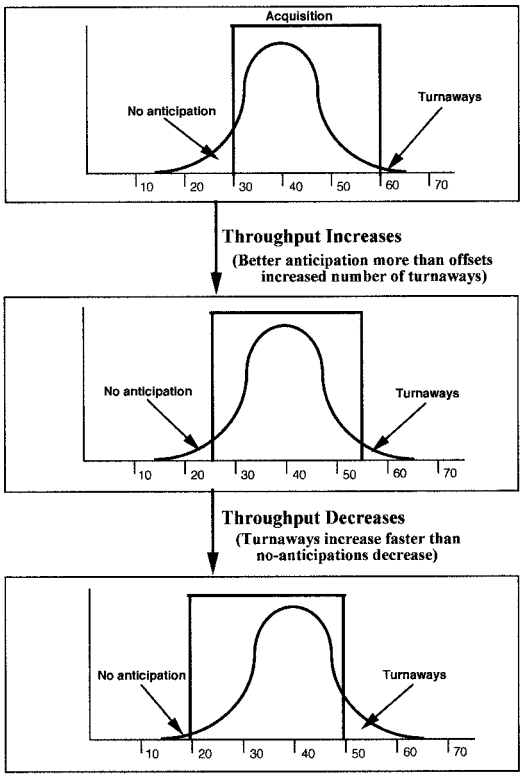

Figure 5: Increasing the amount of time by which we anticipate service completions does not necessarily increase speed (throughput). A little anticipation produces a few turnaways but reduces the time between end of service and next service; with too much anticipation, many of the acquired customers arrive too soon and are turned away, and the acquisition attempt must start over from that point, increasing the idle time between services.

that server, anticipating that this unusually long "service" would end any second now. Some sites had different distributions of durations of service from any we had seen before, requiring a quick change in the file that fed initial estimates to the pacing routines. Once I had worked out the new logic, however, we simply shipped or electronically transmitted the new code, substituted the routine into the control program, recompiled (this took about 10 minutes), and ran. The updated version worked seamlessly on the first try every time.

We installed our first beta test version in early 1987, after about five months of development. Within three months, it was fully operational at all our sites.

Validation

I continued to check the operation of the method in practice and to try to verify its properties theoretically. Most of our sites obtained approximately the results I had predicted: talk times of 50 to 55 minutes per hour (83 to 90-plus percent utilization) with fewer than three percent abandoned calls. The sites whose performance diverged from our predictions had problems other than the pacing logic: highly variable durations of dialing attempts (mostly because of low-quality long-distance carriers), noisy lines that caused problems with recognizing whether the call had been answered, equipment failures, and operators taking breaks without logging off. After much additional work using Markov renewal theory, I derived expected steady-state results for a few cases, and the simulation model's performance tracked the theoretical predictions [Samuelson 1990].

It turned out, in addition, that the policies generated by this method applied to inbound routing and scheduling were in no case inferior to those generated by SMDP. For cases in which the SMDP-generated policies were provably optimal, we obtained the same policies. (These cases involved memoryless arrivals and hence resulted in control-limit policies: The only useful decision variable was the number of customers to allow in the system at one time.) My method reduced to the same

control limits for these cases, admitting new customers when, in an outbound system, one would have wanted to acquire them; where the arrival process was such that the recent history contained useful information, my method exploited it by taking into account, as well, the timing of new admissions.

Assessment of Impact

This work, for which I received a US patent in 1989 (I applied in 1987, just before the first installation), immediately became a key component of ITC's marketing advantage. Both ITC [Varney 1988] and independent reviewers [Kerins 1989] soon came to consider this feature very important and valuable. This type of pacing became and has remained a required feature of systems of this type; a few large competitors who couldn't keep up went out of business. The president of one of these large competitors told me a few years afterward, "For six months after you introduced it, that pacing thing was all any prospect wanted to hear about."

The pacing feature contributed to a large extent to ITC's success and growth and ultimately to its purchase by a competitor, EIS International, which has continued research and development in pacing methods. In 1994, during patent litigation between EIS and a competitor, EIS's legal counsel estimated the value of the patent alone as between $1 million and $2 million. Its value in promoting ITC's and EIS's market position was most likely several times that, but this cannot be measured precisely, as many other factors also contributed to ITC's success.

Summary and Conclusion

ITC's use of queuing and simulation to decide when to dial was a major breakthrough in the outbound telephone-dialing-systems industry. Applying the principle of just-in-time synchronization, implemented via a real-time system that quickly and rather smoothly adjusted to changing circumstances, ITC was able to boost servers' utilization from around 65 percent to 85 to 90 percent, while limiting or reducing the proportion of calls that had to be abandoned. In the process, I obtained the first patent in the US system based on queuing theory and defined and to some extent delineated a new class of queuing models, in which customers do not arrive purely at random but rather in response to a system-initiated acquisition attempt; these models have many other applications. The improvement in performance was readily obvious and highly significant, and the market rewarded the innovation handsomely.

APPENDIX

Formal Statement of the Problem

We calculate

$\bar{S}$ = estimated duration of a service at Q,

$\bar{p}$ = expected proportion of attempts that result in connects (customers proceeding to the service facility)

= P {answer | dial attempt} for the telephone system,

$\bar{A}$ = estimated duration of acquisition (time from dial start to answer, if answered)

and count the number of servers already available and idle, plus the number of servers currently busy but anticipated to be available by the time a new dialing attempt could provide a customer. Anticipation of a service completion occurs, for the ith server at facility Q, when

$\bar{S} - s_i \leq \bar{A}$, where s_i is the time the customer being served by the ith server has

already been in service. When the two sides of this inequality become equal, we expect the ith server to be available to serve the customer we then start to acquire.

Solution for a Simple Analytical Case

While most of the interesting cases were analytically intractable, I was able to obtain steady-state solutions for some simplified conditions, thereby verifying the accuracy and essential completeness of the simulation model. Consider, for example, a single-server system in which service time S has probability distribution $f(S)$ and acquisition attempts whose duration is a constant a and whose probability of success is p.

For such a system, define an anticipative policy (s_β, s^*, m) as simultaneously initiating a set of m acquisitions whenever the server becomes idle with no acquisitions in progress, or when the service in progress has lasted more than s^*, where $s^* = s_\beta - a$ and s_β is chosen such that $P\{S > s_\beta\} = \beta$; and, when all acquisitions in a set fail, initiating another set of m simultaneous acquisition attempts immediately after the last failure.

Denote the number of customers served in time interval $(0, t)$ as $N_S(0,t)$ and the number turned away as $N_N(0,t)$. As I will show, for the cases of interest in this study, steady state exists, and there are three important steady-state measures:
(1) $\bar{R}$, the average throughput (customers served per unit time);
(2) $\bar{U}$, the average proportion of the time servers are busy;
(3) $\bar{B} = E\{N_N / (N_S + N_N)\}$, the expected proportion of customers successfully acquired but then turned away when seeking service.

Define the nth service-idle epoch as beginning when the nth service begins, and let $q = 1 - p$. Finally, we note that sets of acquisition attempts continue until the first such attempt that succeeds after service ends. We must therefore condition on when service ends, then examine the acquisition attempts that conclude—successfully or not—after that. I also denote by I the idle time in a service-idle epoch, so $\bar{R} = 1 / (\bar{S} + \bar{I})$ and $\bar{U} = \bar{S} / (\bar{S} + \bar{I})$.

We now have a theorem:
For a single-server system as defined above, where the service duration distribution, $f(S)$, is continuous and bounded above, and the duration of acquisition $A = a$ (constant, deterministic), utilizing an anticipative policy (s_β, s^*, m), where $s_\beta - a \geqslant 0$, steady state exists, and

(a) $E\{I\} = P\{S < s^*\}\, a / (1 - q^m) + P\{s^* \leq S < s_\beta\}\, [E\{S \mid s^* \leq S < s_\beta\} - s^* + a\, q^m / (1 - q^m)] + P\{s_\beta \leq S\}\, [a / (1 - q^m) - (E\{S \mid s_\beta \leq S\} - s_\beta)]$, and

(b) $E\{B\} = E\{Z\} / (1 + E\{Z\})$, where $E\{Z\} = mp / (1 - q^m) + \beta mp - 1$.

Proof:
These assumptions are sufficient to ensure the existence of steady state, using the key renewal theorem and its corollaries (see, for example, Ross [1970, p. 42 and pp. 95–98] and Samuelson [1990, pp. 32–34 and 79–82] and noting that the service-idle epochs form a renewal-reward process with the starts of service as regeneration points).

(a) The expected time from the start of a set of acquisition attempts to the next successful acquisition is the sum

$a\ P$ {at least one of the first m attempts succeeds}
$+\ 2a\ P$ {all m of the first attempts fail, at least one of the second m succeeds} $+ \ldots = a(1 - q^m) + 2a\, q^m(1 - q^m) + 3a\, q^{2m}(1 - q^m) + \ldots + n\, a\, q^{mn}(1 - q^m) + \ldots = a(1 - q^m) \sum_{n=1}^{\infty} n\, (q^m)^{n-1} = a / (1 - q^m)$.

If the service ends before the start of acquisition, then the idle time is simply the amount of time required to acquire the next customer. If the service time ends between the start of the acquisition attempts and the "target" time s_β, then the idle time for this service-idle epoch is reduced by the amount of time by which service continued past s^*; but if all of the first set of acquisition attempts fail (which happens with probability q^m), then $a / (1 - q^m)$, the amount of time required to acquire a new customer, starting from the time when the first set of attempts fail, is added in as well. If the service continues past $s\beta$, the first set of acquisition attempts finishes too early to enter service; the customers acquired in this set are turned away; and the idle time is the duration of a new set of acquisition attempts minus the time by which the service continued past s_β. With this in mind, we see that the equation is straightforward, as is the derivation of average throughput and utilization from $E\{I\}$.

(b) For each set of m acquisition attempts, mp succeed; sometimes all m fail, however, necessitating another set. Hence the expected value of the number of successful acquisitions, given that at least one attempt in the set succeeds, is

$$mp / (1 - P\{\text{all attempts fail}\}) = mp / (1 - q^m).$$

In addition, we will lose the entire first set of acquired customers if the service in progress at the start of the service-idle epoch continues past s_β. This occurs with probability β and results in the additional loss of the expected number of customers acquired in the first set of acquisition attempts, namely βmp.

Now, exactly one of these acquired customers will enter service, commencing a new service-idle epoch, so the number turned away is the number acquired minus one, that is, $mp/(1 - q^m) + \beta mp - 1$. This completes the proof.

Additional Theoretical Questions

The result extends to more general single-server cases [Samuelson 1990, chapter 4], but considering variable duration of acquisition attempts makes things much more complicated. The straightforward formula in (b) becomes a sum involving conditional probabilities, order statistics, and convolution integrals. Verifying this more general result (it is not quite correct as stated in my 1990 dissertation, but a corrected version in under review for publication) and finding whatever additional generalizations may be possible is a promising area for additional theoretical work.

Other interesting topics include analysis, if possible, of multiple-server systems; systems in which a called party may, after speaking with one server, elect to listen to a recorded message and then become available for another server whose service will be of different (presumably longer) duration than the first server's; systems with multiple campaigns running simultaneously, with the possibility of some servers being reassigned from one campaign to another in real time; and scheduling and routing in inbound and combined inbound-outbound systems.

Slowing Down by Speeding Up

For the purpose of verifying the surprising result concerning the effect of anticipation, the simple theorem proved here is sufficient, as one can readily verify by considering the case with $m = 1$, $p = 1$, service times distributed uniformly from $S_{\min}$ to $S_{\max}$ seconds, and an a-second duration of acquisition. For a uniform distribution $(S_{\min}, S_{\max})$, $s_\beta = (1 - \beta) S_{\min} + \beta S_{\max}$, $P\{S \le s^*\} = 0$, and (substituting into the theorem)

$$E\{I\} = 0 + (1 - \beta)[s_\beta - (S_{\min} + s_\beta)/2 + 0] + \beta(a - [(S_{\max} + s_\beta)/2 - s_\beta]) = s_\beta - \beta s_\beta - S_{\min}/2 + \beta S_{\min}/2 - s_\beta/2 + \beta s_\beta/2 + \beta a - \beta S_{\max}/2 - \beta s_\beta/2 + \beta s_\beta = s_\beta/2 - S_{\min}/2 - \beta(S_{\max} - S_{\min})/2 + \beta a = [(1 - \beta) S_{\min} + \beta S_{\max}]/2 - S_{\min}/2$$

$$- \beta (S_{max} - S_{min})/2 + \beta a = (S_{max} - S_{min})/2 - \beta (S_{max} - S_{min}) + \beta a,$$

whose minimum, clearly, occurs at $\beta = 0$, where we expect the idle time to be half the range of the service time. So the optimum (maximum throughput) occurs at $\beta = 0$ for uniformly distributed duration of service, $m = 1$, $p = 1$, and constant duration of acquisition attempts, as long as the duration of the acquisition attempt is greater than the difference between the maximum and minimum durations of service.

REFERENCES

Kerins, Jack 1989, "Anticipatory dialing: The competitive edge," *Journal of the American Telemarketing Association*, Vol. 5, No. 8 (August), pp. 22–23, 29.

Puterman, Martin L. 1994, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley and Sons, New York.

Ross, Sheldon M. 1970, *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco, California.

Samuelson, Douglas A. 1989, "System for regulating arrivals of customers to servers," US Patent No. 4,858,120, August 15, 1989.

Samuelson, Douglas A. 1990, "Controlling queueing systems with acquisition, DSc dissertation, George Washington University, Washington, DC.

Stidham, Shaler, Jr. 1986, "Scheduling, routing, and flow control in stochastic networks," Technical Report No. UNC/ORSA/TR-86/22, University of North Carolina, Chapel Hill.

Varney, Robert C. 1988, "Measuring outbound calling system performance," *Telemarketing*, Vol. 7, No. 3 (September), pp. 82–85.

F. A. Rhine Morgan III, Vice-President, EIS International, 555 Herndon Parkway, Herndon, Virginia 20170, writes:

"It is my pleasure to verify Doug Samuelson's claim that his method for "pacing" automated telephone dialing systems was successful in practice. In 1987, I was senior vice-president of International Telesystems Corporation, which became part of EIS International in 1993. In that capacity, I was directly responsible for assessing the quality and significance of Mr. Samuelson's work.

"His new method involved a level of mathematical and computational sophistication which, frankly, the rest of us were not prepared to evaluate. Neither we nor anyone else we were willing to consult with on this sensitive technology had ever seen such a use of queueing theory before. We and our competitors were using much simpler pacing methods, generally based on setting a fixed interval between dialing attempts. These methods generally kept operators busy about 35 to 40 minutes per hour, or less, and generated between five and 10 percent abandoned calls. In addition to their direct cost, abandoned calls generate ill will and limit the market for systems such as ours.

"Mr. Samuelson's improved method surprised us first of all by working immediately, and then by the dramatic difference it made in system performance. Most users were able to keep operators busy over 50 minutes per hour, and in some cases nearly 55 minutes per hour, with most experiencing fewer than two percent abandoned calls. "Smart-Pace," as we called it, became the most talked-about development in the industry for about a year after its introduction, as our competitors scrambled to catch up. Most never did.

"As experience led to refinements in the algorithm, we got another pleasant surprise: Mr. Samuelson's insistence on developing the code as a "plug-in" to our control code and debugging it on a carefully constructed "test bed" system, ideas years

ahead of much of the rest of the commercial software development industry, enabled us to implement improved versions with no errors in the field.

"It is difficult to estimate the economic benefit we derived from his invention. In 1994, when we became involved in some patent litigation with a major competitor, our legal counsel estimated that Mr. Samuelson's patent—which opened the class of queueing patents in the US system—was worth at least $1 million, and more likely nearly $2 million, just as intellectual property. In terms of its effect on our business, I believe it would be reasonable to say its value might be as much as 10 times that."