Service Level Notes with Brad Cleveland

The Great Debate: Erlang C or Computer Simulation?

Who's going to win the next Super Bowl? Will Netscape or Microsoft win the browser war? What should you use to predict the agents you need, Erlang C or Computer simulation?... Hey, don't laugh! If you are involved in workforce planning, that's a big question these days.

Ultimately, the debate is centered around how to "get the right people in the right places at the right times." Consequently, the issue impacts service levels, budgeting, costs, agent occupancy and customer satisfaction. Worth looking into, to say the least.

Erlang C — Old Faithful

The widely used Erlang C formula (*see Figure 1*) was developed in 1917 by A.K. Erlang, a Danish engineer with the Copenhagen Telephone Company. Erlang C can be used to determine resources in just about any situation where people might wait in queue for service — whether at a ticket counter, a bank or toilets in a stadium. Erlang C is widely available in the form of free or low-cost PC-based "calculators," and is currently built into virtually all of the full-blown workforce management software packages.

Erlang C calculates predicted waiting times (delay) based on three things: the number of servers (i.e., reps); the number of people waiting to be served (i.e., callers); and the average amount of time it takes to serve each person. It can also predict the resources required to keep waiting times within targeted limits — that's why it is useful for incoming call centers.

As with any mathematical formula, Erlang C has built-in assumptions that don't perfectly reflect real-world circumstances. For one, it assumes that "lost calls are delayed." In plain English, that means that the formula assumes that calls are queued. No problem with that. The problem is that it assumes callers queue as long as it takes to get an answer, or that nobody will abandon. *Oops!* Erlang C also assumes that you have infinite trunking and system capacity and that nobody will get a busy signal. But some call centers have quite a problem with busy signals. *Oops again!*

Figure 1

$$P(>0) = \frac{\dfrac{A^N}{N!}\,\dfrac{N}{N-A}}{\displaystyle\sum_{x=0}^{N-1} \frac{A^x}{x!} + \frac{A^N}{N!}\,\frac{N}{N-A}}$$

Where
A = total traffic offered in erlangs
N = number of servers in a full availability group
P (>0) = probability of delay greater than 0
P = probability of loss — Poisson formula

The result is, in a nutshell, Erlang C may overestimate the staff you really need. If some of your callers abandon or get busy signals, your reps won't have to handle all of the calls Erlang C is including in its calculations. For a given level of staff, Erlang C predicts that conditions will be worse than they really are. Erlang C also assumes you have the same level of staff on the phones the entire half-hour. In reality, if service level starts taking a nose-dive, you may be able to add reinforcements on short notice.

So, just how bad is Erlang C, anyway? "Erlang C is fairly accurate for good service levels," says Mike Hills, a software developer and recognized expert in traffic engineering. "However, for poor service levels, Erlang C overestimates how bad it really is. Reality will be nowhere as bad as Erlang C predicts."

So why is Erlang C so popular? As you might guess, there are defen-

Table 1

Advantages of Erlang C	Disadvantages of Erlang C
· Assumes random call arrival and that calls queue if a rep is not immediately available.	· Assumes no abandoned calls or busy signals.
· Is accurate at good service levels, where abandoned calls and busy signals are minimal.	· Assumes "steady state" arrival, or that traffic does not increase or decrease beyond random fluctuation within the time period.
· Is easy and quick to use and available in software form from a wide variety of sources.	· Assumes you have a fixed number of staff handling calls throughout the time period.
· Is the basis for staffing calculations in almost all workforce management software programs.	· Assumes that all agents within a group can handle the calls presented to the group.

Table 2

Advantages of Computer Simulation	Disadvantages of Computer Simulation
· Can be programmed to assume a wide variety of variables, such as overflow, overlapping groups and skill-based routing.	· Takes time to set up and use, and requires a relatively advanced user.
· The assumptions can include lost calls and busy signals.	· Is a stand-alone tool that is generally not integrated with forecasting and staffing modules.
· May be programmed to use the terminology of your ACD vendor, for ready translation into your environment.	· Is generally more expensive than stand-alone Erlang C programs.

sible reasons to use it. For one, it's a planning tool, and most call centers are *planning* to have good service levels. When service level is decent, you should theoretically have little in the way of lost calls or busy signals. If you *do* have a lot of calls disappearing or getting busy signals, it's probably because you don't have enough staff to handle the load. In that case, who's worried about overstaffing? As your staffing more accurately reflects the workload demand, Erlang C will inherently become more accurate.

Further, if you try to adjust for abandoned calls and busy signals, and retry rates are higher than you estimate, you could end up underestimating staff. (And frankly there's a little industry secret... some call center managers have decided that a little over-calculation as a safety net isn't such a bad thing. They figure that they fail to get full effective use of their already authorized headcount anyway, due to staff turnover and the time it takes to hire and train replacements.)

Currently, Erlang C is still predominant in workforce management software. It is designed for straight-forward environments, like sales calls going here and customer service calls going there. Says Jim Oberhelman of simulator-provider Bard Technologies, "Erlang C was quick, easy and good enough, until ACD and network providers introduced complex routing capabilities."

Oberhelman hits the nail on the head. The realities of today are not as straight-forward as they used to be. Elaborate routing contingencies, such as agent groups that overlap, skill-based routing and complex network inter-flow are common examples. In these applications, Erlang C is "kludge" at best and totally unworkable at worst.

Computer Simulation — New Kid on the Block

Enter call-by-call computer simulation. These simulators do for call centers what flight simulators do for pilots or aircraft designers – they enable you to test your staffing and system programming assumptions before you actually implement changes. Consequently, simulation program providers (e.g., Bard Technologies, Systems Modeling Corporation, TCS, Rockwell and others) are riding a wave. All indications are that there will be increasing demand for these packages.

However, computer simulation also has some downsides. First, simulation is designed for modeling, design and verification, and is generally not meant to be a forecasting and scheduling tool (neither is Erlang C, but Erlang C is often integrated with forecasting and scheduling modules in workforce management systems). It's usually provided as a stand-alone system (although Rockwell's simulator is built into their ACD). "What people have to realize is that simulators provide a way to test ideas about changing your configuration or the way that calls are handled, before introducing the changes," says USAA's Terry Trevino (see *SLN*, September 1995, page 1). You will still need your forecasting and scheduling software.

Second, simulation software takes a lot more time to set up and use than Erlang C. Like a flight simulator, you have to run it over and over to identify potential results. That is a phenomenon of its added flexibility, and the time spent will be time saved if you have a complex environment that requires a simulator's perspective. But it takes time to feed variables into the program and interpret the results.

And the Winner Is...

So what should *you* use? For fairly straight-forward environments with good service levels, Erlang C remains an accurate tool. But if you are utilizing complex routing capabilities, we recommend that you use simulation to validate your system configurations and staffing plans. There is something to be said for a combination of Erlang C, intuition and experience, but simulation will obviate the need for a lot of guesswork.

In many cases, the ideal solution is a combination of both methodologies. In fact, most of the call centers using simulation today continue to use Erlang C for routine staffing and budgeting. Computer simulation is a much more powerful tool for analyzing specific complex scenarios. But Erlang C remains an excellent, easy-to-use tool for illustrating call center dynamics (i.e., when service level goes up, occupancy goes down), and is predominant in workforce management software.

Whatever methodology you use, remember that no formula or program can perfectly predict the future. As Hills says, "As much as I love it, traffic engineering is only a guide — not omnipotent." SLN