

# Call Queue Dynamics

Improve your understanding of the relationships between workload, service-level, abandoned calls, and staff levels.

by Michael Tanner, FIMA CMath

Copyright © Mitan Ltd. 1998-2004

Managing a call-centre requires a variety of skills. Dealing with people is perhaps the most obvious, plus knowledge of the business and market served by the call-centre. You need some understanding of IT and communication technology. Another essential and quite separate skill is an understanding of queue dynamics. This article is intended to help you understand some of the basics of queue dynamics.

**Why Not?** Some topics have a suggested action you could take to apply the points made to your own call-centre. Suggestions are highlighted like this.

## Stress

Queue dynamics involves probability, chance, and statistics. Most people have difficulty in reasoning clearly about chance or probability. There are many examples of this, ranging from gambling to how juries assess some types of evidence. The results are steady profits for casinos and bookmakers, and sometimes miscarriages of justice. In a call-centre a lack of understanding of queue dynamics can mean failure to meet targets for performance, and added stress at all levels. Stress can arise from unrealistic or conflicting performance targets, or just a feeling of being out of control.

## Where To Start

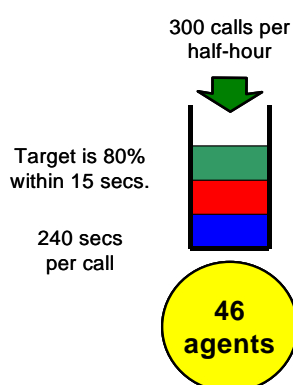


Figure 1. Example

To explain queue dynamics we start with a simple situation of one queue and one group of agents. More complicated arrangements with multiple queues and several agent group will need more sophisticated analysis methods than those used for this article, but the concepts presented here will still apply.

Fig. 1 shows the example we shall use. A single agent group serves a single queue of calls. The call rate is 300 calls per half-hour, calls last on average 240 seconds, and the service-level target is to answer 80% of calls within 15 seconds.

## Develop Your Intuition

If you want to understand call centre performance then it's important to develop an "intuitive feel" for queue dynamics. Although the complicated maths involved in queueing calculations can be hidden in computer programs, don't make the mistake of thinking that, since you have a call-centre planning package, you don't need to understand queue dynamics. Planning programs can organise the data for you and do the complex calculations, but you still need to understand the nature of the decisions you are making about staffing and performance.

## Bringing Queue Dynamics Alive

To get the most from this article and start to develop a good intuitive feel for queue dynamics, you should do the exercises, although the article is designed so you can just read it and still get the main messages. The exercises require a copy of the MITAN PhoneCalc program, which you can get at no charge. (See panel at end of article). PhoneCalc is an advanced call-centre calculator with an intuitive interface. With PhoneCalc you can work through the exercises and experiment with your own workload and agent numbers. PhoneCalc will help bring queue dynamics alive for you, and has been very successful as an integral part of training programmes developed jointly by MITAN Ltd. and Grafton Consulting.

**PhoneCalc.** Exercises requiring PhoneCalc are displayed like this. If you haven't got a copy of PhoneCalc then just skip over these paragraphs.

**Why Not?** Go and get a copy of PhoneCalc. If you're not familiar with the internet, then find a colleague who is, and get them to download PhoneCalc from [www.phonecalc.com](http://www.phonecalc.com).

## The Erlang-C Triangle

The basic call-centre queueing formula is called Erlang-C, named after A.K. Erlang who pioneered the analysis of queues about 1910-20. Erlang-C explains the relationship between the workload (call rate and call duration), the number of agents, and the service level. Fig. 2 shows the way we normally use Erlang-C in a call-centre, where we know the workload forecast for a particular half-hour interval, and we want to calculate the number of agents we need.

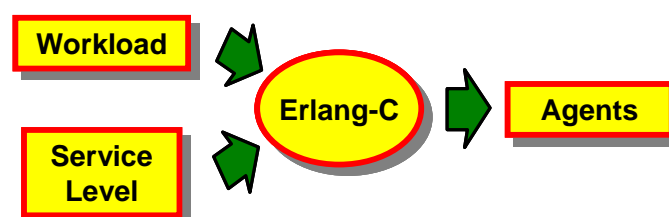


Figure 2. Calculating agents needed

Taking the example in Figure 1, Erlang-C tells us that we would need 46 agents to achieve the service level target. The calculations also tell us other things, such as the agent availability is 13%, the average queue size is 1.8 calls, and the actual service level is 81.7%.

**PhoneCalc.** Set 300 calls per half-hour and 240 seconds call duration in the panel headed "Specify Workload", and set the service-level target of 80% within 15 seconds in the panel headed "Targets and Limits". Finally, look at the panel headed "Set Number of Agents" and check that "By Erlang-C" has been selected. Take a few minutes to find the various results on the "Summary" diagram. There is an extensive help file to explain the various statistics displayed.

## Why Is Agent "Idle" Time Necessary?

In the example agent availability is 13%? Why does this have to be so high? (Remember, this 13% doesn't include breaks or off-phone tasks.) The reason is that calls arrive randomly and bunch up, with sometimes lengthy gaps between calls. Fig. 3 shows a simulation of call arrivals. The call rate is 300 calls per half-hour, and the chart shows an hour of activity. (A constant call-rate has been used for simplicity.) On average 10 calls will arrive each minute, but in some minutes there are many more and in others many less. Since only short waits can be tolerated, calls arriving during peak minutes cannot all be kept waiting to be handled during "trough" minutes. During the troughs there are bound to be some unoccupied agents. The longer the acceptable waiting time, the more the peaks can be held in the queue to fill up the troughs.

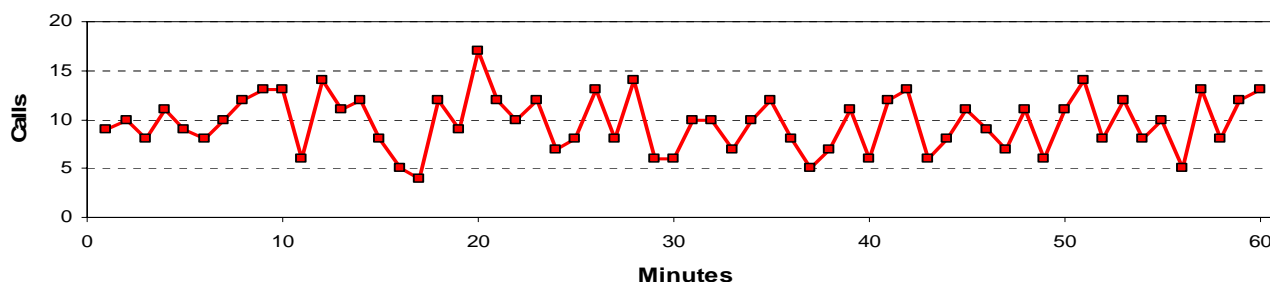


Figure 3. Calls arriving minute by minute

## Productivity and Service-Level

Several jargon words are used in connection with how busy the agents are. "Availability" means the percentage of time the agents are unoccupied while available to take calls. "Occupancy" is just the opposite of this, and is the percentage of time agents are busy with calls. So availability plus occupancy is 100%. Note that breaks and other off-phone activities are excluded from these calculations. It is tempting to regard "occupancy" as meaning "productivity". But the reason agents cannot be 100% busy is the randomness of call arrivals, not a lack of efficiency in the way agents work.

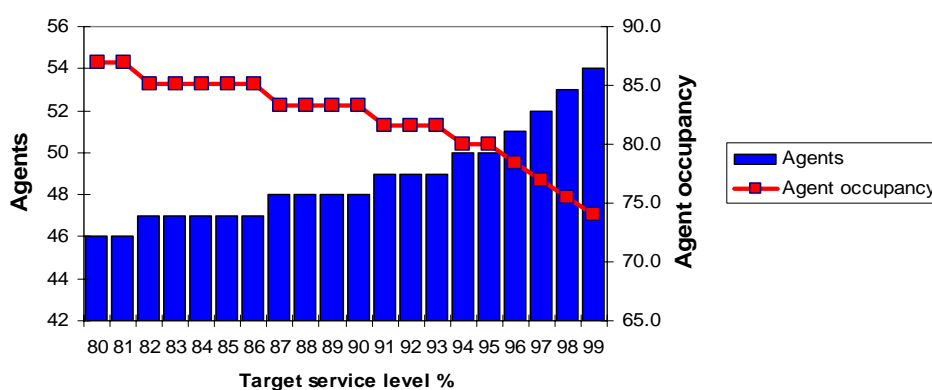


Figure 4. Occupancy and service level

The important link between occupancy and service level is illustrated in Fig. 4. As the service-level target is improved by increasing the percentage of calls to be answered within the target time, more agents are needed, and so the occupancy becomes lower.

**Why Not?** Check over some recent ACD reports to see whether your occupancy is about right for your service level. Be careful how you calculate occupancy, it is the percentage of "plugged-in" time that is spent answering calls or in wrap.

## Economies of Scale

Large groups of agents can achieve a given service level at higher occupancy than a smaller group. This is illustrated in Fig. 5. As the call rate is increased, we need a larger group of agents, but the number of agents needed goes up more slowly than the call rate. The economy of scale effect is more marked at the small end of the scale.

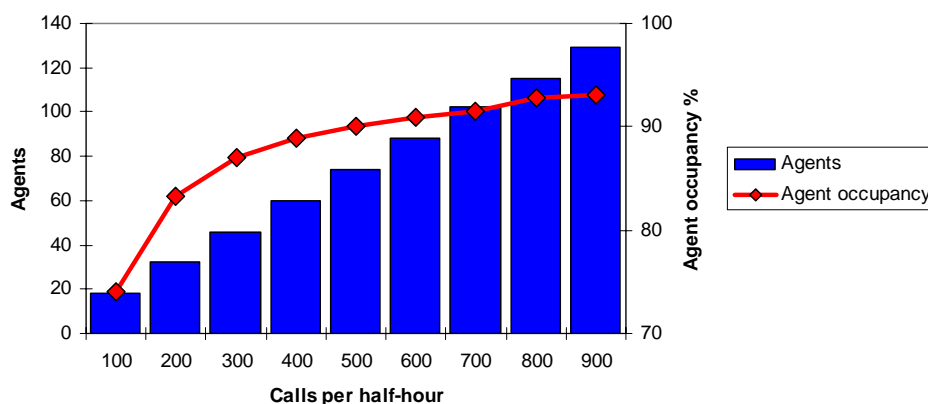


Figure 5. Economies of scale

Of course, you may have a huge call-centre, but be operating in small groups. Conversely, you may have a number of small locations that are effectively networked into a large group.

**PhoneCalc.** Start with the standard example and look at the occupancy or availability. Now change the call-rate, allowing PhoneCalc to set the number of agents by Erlang-C. Look at how the number of agents and the occupancy change.

## The High Cost of Not Enough Agents

Erlang-C tells us that in our example we need 46 agents to meet the service-level target. Let's look at the effect of assigning either fewer or more agents. This time we shall use Erlang-C as shown in Fig. 6, where workload and agents are specified and we need to calculate the resulting service level.

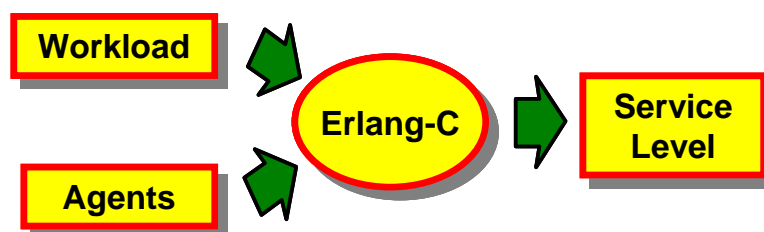


Figure 6. Calculating service level

**PhoneCalc.** Check that workload and targets are set to the example values. Select "Manually" on the "Set Number of Agents" panel, and then experiment with changing the number of agents via the field at the top of the panel. You can also select "Other Graphs", then "Performance vs. Agents" to see a graph like Fig 7.

The service level for different numbers of agents is shown in Fig 7. Starting with 46 agents we get 81.7% service level. With 45 agents we get only 75.1% service level, a reduction of 6.6%. Going to 44 agents gives 66.4% service level, a further reduction of 8.7%, and so on. We would obviously expect the service level to deteriorate as we remove agents, but the deterioration is bigger for each agent removed. Removing 4 agents out of 46 halves the service level from 82% to 41%!

If we add agents we get less benefit from each agent added. Going from 46 to 47 agents improves service level from 81.7% to 86.8%, a gain of 5.1%. Adding another to get 48 agents gains us 3.7% on the service level and so on.

Managers might note there are big performance penalties for being even slightly under-staffed, or allowing agents to be sidetracked with non-phone tasks, and not much to gain from over-staffing.

For agents the message is that each individual really matters. It might seem that one person out of 46 isn't important, but one agent back late from a break, or not taking calls when they should be, can have a big impact on the service level.

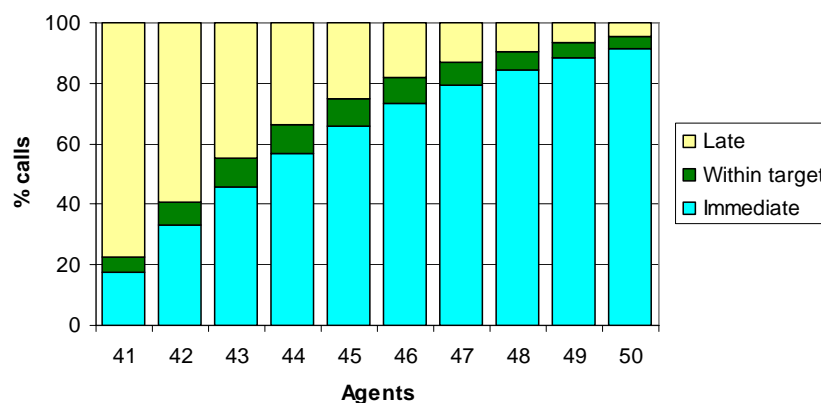


Figure 7. Service level versus agents

**Why Not?** Create a simple chart similar to Fig. 7 to show your agents how much each individual matters to the overall performance. You can use PhoneCalc with your own data to print out a chart. Emphasise management responsibility to schedule enough agents, and the agent's responsibility to be available for calls.

### How Long Do Calls Wait?

The average wait time, usually called the average speed of answer or ASA, is often used as an overall measure of performance. For our example, the ASA is 10.6 seconds. Since ASA is over all calls, it is sometimes mistakenly assumed. In order to achieve typical service level targets, a large proportion of the calls have to actually be answered immediately. This can be seen in Fig. 7. For our example, 73% of calls will be answered immediately. Now if nearly three quarters of calls have zero wait, and the ASA is 10.6 seconds, then the other quarter of calls must be waiting longer than 10.6 seconds.

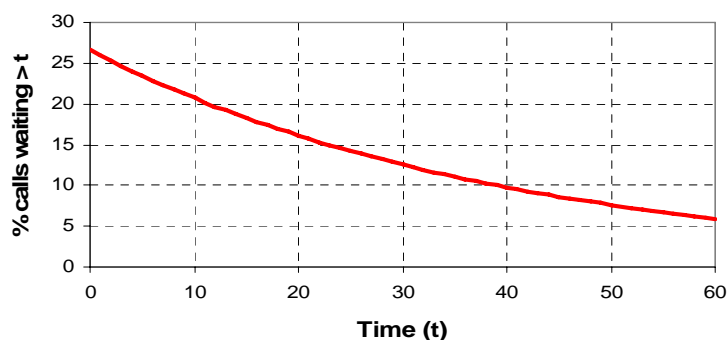


Figure 8. How long do calls wait?

We know from the service level that 18.3% of calls (100-81.7) will wait longer than 15 seconds. We can use Erlang-C to draw Fig. 8, which shows how many calls wait longer than any given time T. In fact, nearly 6% of calls wait longer than a minute! The average wait for calls not answered immediately is 40 seconds. Looked at this way the service level of 81.7% within 15 seconds doesn't seem so impressive. This way of looking at service levels explains why you may see significant abandon call rates at the same time as achieving good service levels.

**PhoneCalc.** Set workload and targets to the example values. Select the "Additional Detail" display and look at the "Calls Answered" chart, which shows how many calls are answered immediately. Next to this is the "Average delayed wait" which is the average wait time for calls not answered immediately.

**Why Not?** Consider your own service-level targets. With your own data and PhoneCalc you can get lots of information to see what your chosen service-level really means. Find out how your targets were decided upon, but remember that many call-centres have little guidance available except what seems reasonable.

### Queue Size - What Does It Mean?

Most call centres have wallboards showing, amongst other things, the queue size. The obvious interpretation is that a big queue size is bad, but is this true? Have another look at Fig. 3. The queue of calls is the mechanism for balancing the short-term peaks with the short-term troughs. So some queueing is essential.

**PhoneCalc.** Start with the standard example and look at the average queue size, which is 1.8 calls. Now change the call-rate, allowing PhoneCalc to set the number of agents by Erlang-C. As the call-rate changes look at the average queue size.

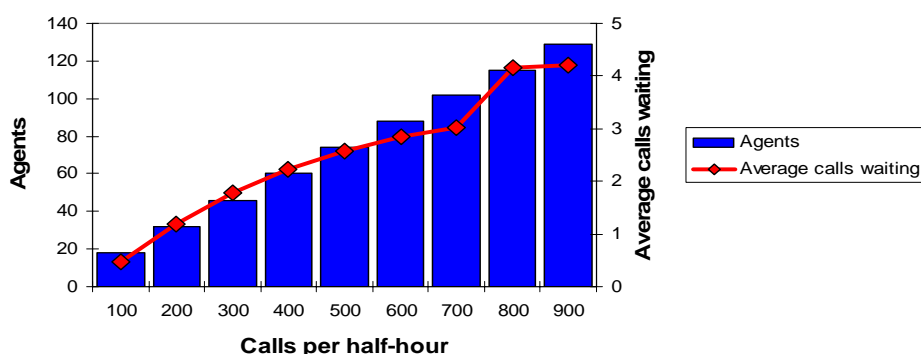


Figure 9. Queue size

When achieving typical service-level targets, a small agent group will have a small average queue. A larger group can have a larger average queue and still be achieving the target.

Fig. 9 shows the average queue size according to the call-rate. For a particular call-rate the number of agents needed for the service level has been assigned. From this you can see that larger the operation, in terms of workload and agents, the larger the average queue size. Since this is an average, you will sometimes see longer queues, but the graph gives an idea of what to expect (The queue size curve is not smooth because we cannot assign part of an agent, so graphs of performance or queue size often have sharp jumps.)

**Why Not?** Use PhoneCalc with your own data to see what the average queue size is predicted to be. Use this as a yardstick for comparing the actual queue size you see in your reports or on the wall-board. Don't forget that the wall-board is showing the queue size as it varies, not the average.

## Abandoned Calls

MITAN has developed a more sophisticated version of Erlang-C that takes into account caller tolerance, and predicts both service level and abandon rate. The approach is illustrated in Fig. 10, which shows the additional factor of "caller tolerance" being used as an input, and abandon rate as an additional output.

Mitan's approach differs fundamentally from the "empirical adjustments" that are sometimes used, and is probably the only method of properly taking into account the subtle and significant interaction between abandoned calls and service level.

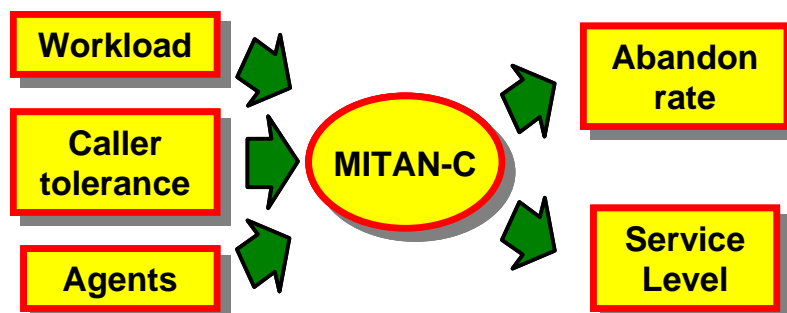


Figure 10. Mitan-C for abandoned-call analysis

PhoneCalc includes Mitan's abandoned call analysis method. You can select it by clicking the "Model" button at the top of the main display. General use of the abandoned call method requires a chargeable licence, but without a licence some examples can be demonstrated. Switch to the "Abandoned Calls" model and explore the results.

## Service Level and Abandon Rate

Fig. 11 shows what happens for our standard example when abandon calls are considered. Fig. 11 can be compared directly with Fig 7. If we assign 46 agents as before, then instead of an 82% service level, we now get a 92% service level.

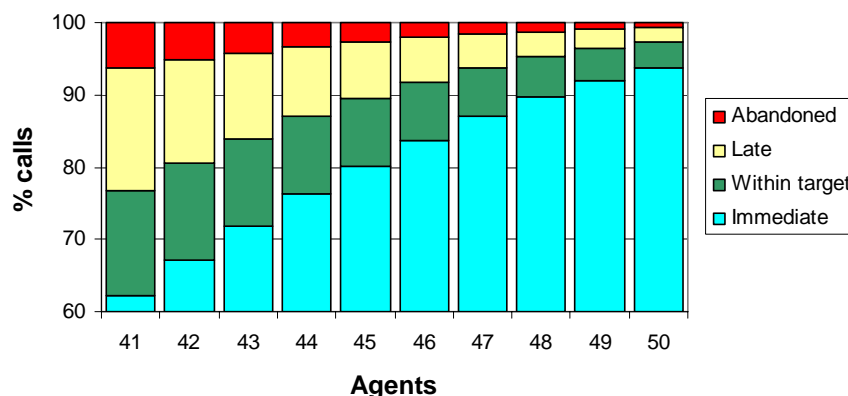


Figure 11. Service level and abandon rate

Call-centres are often puzzled that quite a lot of calls abandon even though the service level seems very good. In fact the service level is good because of the abandon rate!

**Why Not?** Look at some of your recent ACD reports and check whether PhoneCalc "explains" the service levels and abandon rates. You may have to adjust "caller tolerance" to improve the fit, although results are not particularly sensitive to this value. (This is the only exercise or suggested action in this article that needs a chargeable licence.)

## Which Target?

From Fig. 11 we also see that the 80% service level target can be achieved with just 42 agents, rather than the 46 required before. This is of course at the cost of having over 5% of calls abandoning. So for each number of agents, we get a particular service level and abandon rate. Most call centres have a target service level and also a target for the abandon rate. With Mitan's abandoned call method you can now plan how to meet both these targets.

As a general rule, it takes more agents to achieve typical abandon rate targets than it does to meet typical service-level targets. This means the kind of service levels that most call-centres think are reasonable are not sufficient to keep abandon rates down to acceptable levels. Possibly it is not widely enough understood that typical service levels mean most calls will be answered immediately, but quite a lot will face a surprisingly long wait. ASA as a measure of performance, should really be avoided.

**Why Not?** Find out how important abandoned calls are, compared to service-level and agent occupancy. Is it possible to put an objective value on a lost call? Discuss whether the nature of your business means that callers are prone to abandon

## Managing Overloads

Suppose we can only assign 46 agents, but are expecting an exceptional peak of 400 calls per half-hour, one-third greater than we have the staff for. According to Erlang-C we would expect disaster. The service level would be zero and the queue would just keep getting bigger (until we ran out of phone lines at least!).

In practice, things would be bad, but not as bad as Erlang-C predicts. ACD reports would show something like the Mitan-C prediction of 45% service level and 16% abandon rate. Foreseen or not, overloads do occur and a planning method that can handle overloads enables you to manage more effectively and to see whether you did as well as you could have in the circumstances - an important piece of feedback to agents.

## Conclusion

Queue dynamics is vital to effective management of call-centre performance. It is important for both junior and senior managers to develop a good intuitive understanding of queueing dynamics if they are to make sound planning decisions and react effectively to day-to-day situations.