

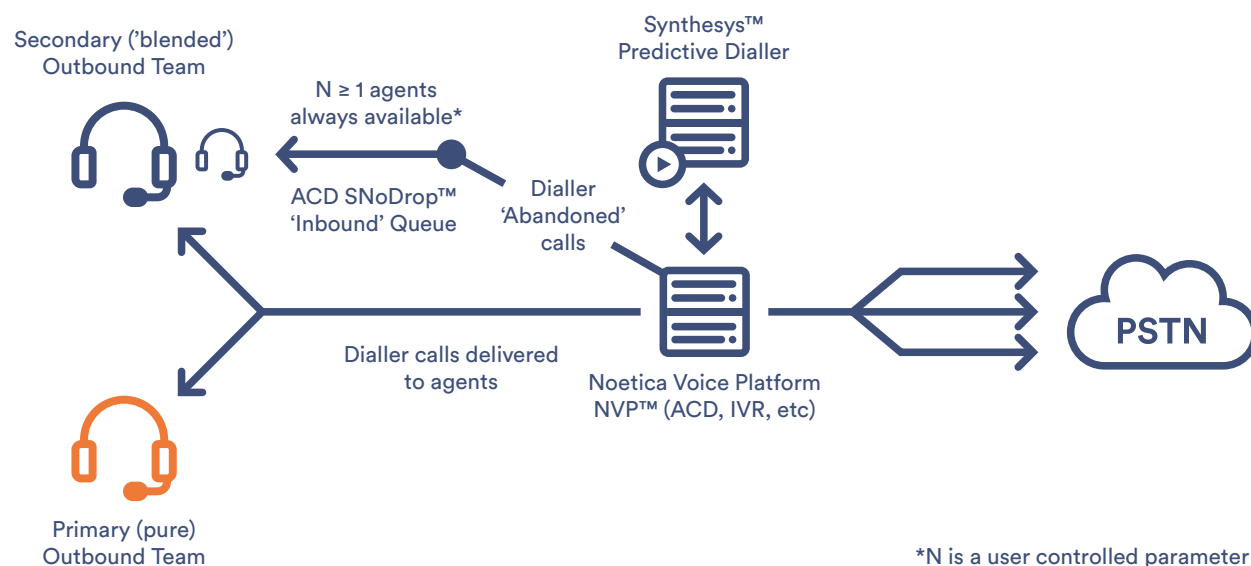
SNODROP™ PREDICTIVE DIALLING WITH NO DROPPED CALLS

All predictive diallers generate 'abandoned' (also known as 'dropped') calls. This is a direct consequence of the non-deterministic nature of the fundamental algorithms which all predictive diallers rely upon. To be precise, dropped calls are largely the result of a probabilistic overestimation of the number of calls a dialler should be making at any point in time in relation to the volume of agents it is required to service and the connection rate of the data in use.

Therefore, the general consensus to date has always been that dropped calls are an inevitable and unavoidable side effect of all and any predictive dialling activity. The only palliative measure on offer up to now was to limit the number of dropped calls to what might be considered 'acceptable' levels by limiting the pace of dialling once an agreed limit is breached.

However, as regulation surrounding the use of predictive dialling and abandoned calls tightens around the world, what might be considered 'acceptable' with regards to tolerance towards dialler dropped calls is drawing ever closer to zero. Under normal circumstances, this should be sounding the death knell for predictive dialling as an acceptable practice. At Noetica we have recognised this trend well before time and have been working closely with our dialler clients to develop new technologies that address these concerns. As a result, we have recently announced our new SNoDrop technology which delivers responsible predictive dialling with virtually zero abandoned calls.

Of course, this may seem a little too good to be true, so here is a high level explanation of this new technology. Fundamentally, it relies on combining our predictive dialler with Noetica's Voice Platform (NVP™) and our call blending technology to eliminate dropped calls or indeed reduce them to a completely negligible level.



The method can be summed up as follows (please refer to the diagram below):

1. On the NVP™, define an ACD inbound queue (known as a 'route') dedicated to SNoDrop™. The SNoDrop™ queue is not intended to route any inbound calls (although the method does not prevent it). The queue is set up with a maximum 2 seconds wait time during which period a call in the queue will either be delivered to an agent or dropped with a message.
2. For each dialler campaign, define two teams: a pure outbound team A and a secondary team B which uses Synthesys™ call blending to accept calls from the dialler and from the SNoDrop™ queue. Team B can be as large as the entire agent contingent (meaning that team A is empty) but cannot be smaller than a minimum calculable number of agents.
3. Synthesys™ blending requires a parameter N ($N \geq 1$) which designates the target number of agents that should be kept available for inbound at any given point in time. For instance, if N is set to 1, a single agent in the blended team will be kept available to handle possible inbound calls. When that agent receives an inbound call, the next outbound agent to become available is taken out of the dialler and becomes the current agent earmarked for inbound duties. Conversely, if more than one agent is seen as waiting for inbound, then one of them will be returned to the dialler.
4. By enabling SNoDrop on the dialler campaign, any call that the dialler would normally drop (abandon) would be placed into the SNoDrop queue and will be delivered to one of the blended agents on the B team who is on 'inbound' duty.
5. The NVP™ will inform the dialler of the final outcome for each call placed in the SNoDrop™ queue. Most if not all of these will reach agents in team B. In all these cases the dialler will deduct one from its abandoned call count. As a result, the abandoned call rate should hover around zero.

The SNoDrop™ technology has been in live trials for some time and is indeed meeting all our expectations. We are confident that SNoDrop™ is a game changer in the industry and proves that it is possible to deliver responsible predictive dialling with zero or near zero abandoned calls.

