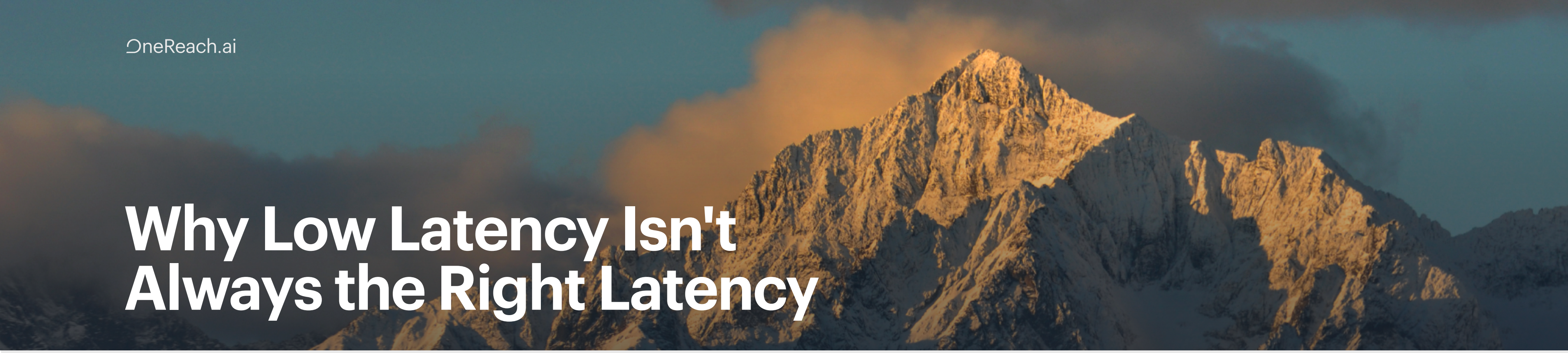


OneReach.ai

Why Low Latency Isn't Always the Right Latency

Designing Successful Conversational Experiences Isn't Always about Speed

For bots and humans alike, successful conversations hinge on the ability to listen well. It's nearly impossible for bad listeners—whether they're humans or machines—to provide excellent customer experience. So while it's often tempting to emphasize the speed with which a machine can reply to a query, the smarter design choice is often to slow things down to make sure a user can finish a thought before the machine jumps toward formulating a response.

One of the key indicators that help bots cater to the way that humans converse is by having them look for a pause—called an **inter-speech time out**. This is reflective of the way humans know if someone has completed a thought, and it's one of the most simple options to look at when working on end of speech detection.

There's a tendency with conversational AI to want to show off a bot's fast response time—essentially parading around algorithms that are speedy when responding to a sentence. While this makes for a flashy demo, if a real life user says "yes" before pausing to complete their thought, the bot is left needing to backtrack and start from an earlier point in the conversation. This scenario creates a puzzle for conversational designers and frustrates users. It should be avoided whenever possible. It's better to be accurate than fast, and you can still give someone feedback so they are not waiting in silence.

It's hard enough for humans to quickly determine that someone has completed a thought—it can be even more challenging for machines, especially over the phone. When they fail at this, machines often cut off a user or completely misunderstand their request.

Good bot experiences require bots that are good at listening, but a bot providing a faster response doesn't necessarily mean it's providing a better experience.

The goal isn't to mimic human behavior, it's to be useful. An interaction with a machine that's mildly stilted because it's being careful to make sure users finish their thoughts is vastly preferable to one that's trying to match the flow of natural conversation, or dazzle with speed.

Inter-speech timeout is a design consideration, not a latency consideration. Good design requires context.

Inter-speech timeouts are critical to the success of conversational AI on voice modals. By definition, inter-speech timeouts are the pauses between words or phrases in speech. The length of these pauses allow your chosen speech to text (STT) engines or your voice engine to determine when a customer is done speaking (or when they complete a thought) and to process results.

Voice platforms and STT vendors rarely allow for the granular control of these timeouts for each user-response. We believe this level of fine tuning is critical for the success of any voice based solution. In order to create a great voice experience, you need granular control of inter-speech timeout by phrase so that you can design for the context of the question. This includes a combination of STT options and direct voice platform control.

There are several key considerations to take into account when optimizing inter-speech timeouts. Here are four.

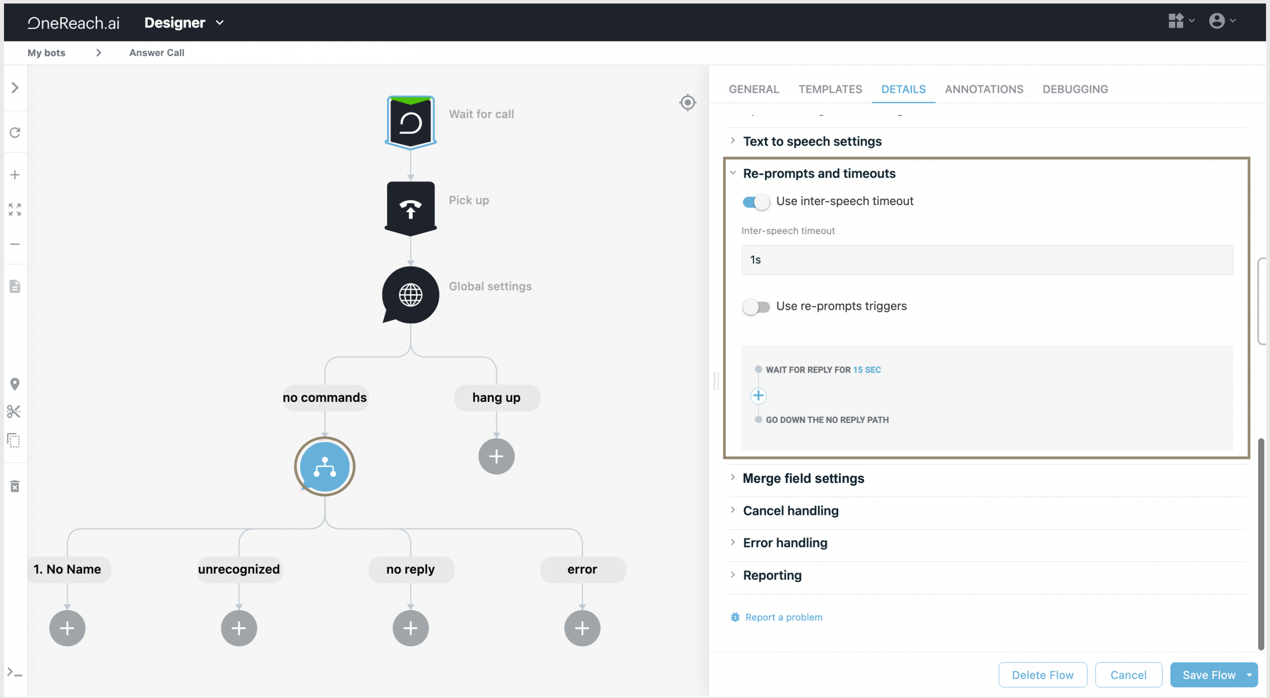
#1 - Create pauses that are long enough to allow the software to correctly identify the words and phrases, but not so long that it becomes disruptive to the conversation

First, the length of the timeout should be long enough to allow the software to correctly identify the words and phrases, but not so long that it becomes disruptive to the conversation. Take for example the scenario of collecting a credit card number. How does the bot know when a user is done speaking?

Imagine yourself sharing your credit card number: "My credit card is 555...718......87922....1". You might struggle to read the numbers or accidentally drop your card while reading, despite being a very fast speaker otherwise.

Too small of an inter-speech timeout can cause the experience to break down. You might have to repeat the number, or go to an agent. Containment and your experience as a user would be shot. Sometimes the necessary latency might seem a little long in terms of the natural flow of conversation, but that extra time might be the difference between high containment rates and low containment rates.

When designed properly, IVRs have shown they can be better at things like collecting credit card numbers. Inter-speech timeouts can be the difference between improving these experiences or making them substantially worse. After the initial pause of “555” the bot would cut off the user and proclaim, "That's not a full credit card number. Let's try again". Too long—30 seconds—and the user might think the bot has disengaged.



Re-prompts and timeouts

● Use inter-speech timeout

Inter-speech timeout

1s

○ Use re-prompts triggers

WAIT FOR REPLY FOR 15 SEC

GO DOWN THE NO REPLY PATH

The importance of inter-speech timeouts lies in the fact that they provide cues for the speech to text software to correctly identify words and phrases while managing the latency of responses.

#2 - Optimize for each response and context

Inter-speech timeouts should be optimized for each type of response and their respective context, on a step-by-step basis. They should never be set globally for an entire conversation. For example, yes/no questions can have shorter timeouts than when asking a user for their child's birth date. If we are trying to transcribe a meeting, we can afford to be more lenient with our inter-speech timeout because the user is not expecting an immediate response. In this case, a longer timeout of maybe 5 seconds may be more appropriate. Waiting in this scenario is not critical to the user experience and thus allows us the flexibility to value accuracy over speed.

#3 - The Acoustic Environment

When managing end-of-speech detection we also need to take into account the acoustic environment. If a user is in a noisy place, like a busy street, the system will need to be more lenient with end-of-speech detection to account for the background noise. Think of taking a call on a factory floor, or in a home with kids in the background. It's easy for the bot to mistake the noise for part of the user's speech and get confused. It's critical to distinguish actual user speech from unrelated noise and deliver the right experience to customers.

#4 - Timeout is a design consideration, not a latency consideration

As with every aspect of conversational design, context is everything, and **it's important to understand how end-of-speech detection affects the perception of latency**. Generally speaking, you have roughly two seconds to determine if a user is done talking. What you do from that point on is a different story. Processing is the time it takes to interpret what the person has said, plus any time it may take to process and confirm the information with any other systems. If a person wants to check on an order, the system may take several seconds to get a response from a legacy system.

Latency Equation

ESD + P = R

Latency is the accumulated amount of time it takes to determine they've completed their thought (End Speech Detection - ESD), plus the amount of time it takes to process their statement (P) and then respond (R). Processing time includes executing internal processes like APIs, NLU, TTS, logic parameters, etc. OneReach.ai average response time (R) is 500 Milliseconds, and industry standard processing time is on average 2-3 seconds.

Legacy systems typically take longer to process and respond. And how you respond should be defined more by contextual information than concern over latency times (see credit card example above).

No matter how long it takes you to process and respond, you need to give the user some kind of feedback. **After about three seconds, users typically need a signal that lets them know that something productive is happening on the other end.** A simple, "one moment please" can make all the difference

Conclusion

Inter-speech timeouts are critical to the success of speech to text for conversational AI. The timeout should be optimized for the task at hand and the acoustic environment. It's a difficult balance to strike, but one that is necessary to provide users with the best possible experience. You need to be sure you're equipping yourself for this art so you can get the most out of your speech-to-text application.

Trying to predict when a person has completed a thought is a party trick with not a lot of upside. Instead, try to optimize for each question and context for the highest containment rates - look to the data around containment rates (in particular, your rate of understanding) as your north star.