Latency

Voice Agent Networking Event

Definition

Latency is the time between user speech end and agent response start. Human conversation responds in 200-300ms. Voice agents exceeding 500ms feel unnatural.

Components

Network RTT adds 20-100ms based on geography. VAD detection requires 100-300ms to confirm speech end. Speech-to-text processing takes 200-500ms. LLM inference ranges 500-2000ms. Text-to-speech synthesis needs 100-400ms. Total pipeline averages 1000-3500ms.

Voice Activity Detection

VAD determines when users finish speaking. Shorter thresholds (300ms) feel responsive but risk interruptions. Longer thresholds (800ms) prevent cutoffs but feel slow. Context-aware VAD adjusts dynamically.

Predictive VAD begins processing before speech ends, reducing perceived latency by 200-300ms. ML models use prosody and speaking rate to predict utterance boundaries.

Speech Recognition

Streaming ASR provides 50-100ms latency during speech versus 200-500ms for batch processing. Streaming enables downstream processing before users finish speaking.

Model size trades accuracy for speed. Large models achieve 95% accuracy in 400-600ms. Small models reach 90% accuracy in 100-200ms. For voice agents, speed often wins.

GPU acceleration reduces ASR latency 5-10x versus CPU. Cloud services offer powerful GPUs but add network delay. On-device processing eliminates network latency but requires smaller models.

Language Models

LLM processing is the largest latency component. GPT-4 takes 1-3s for first token, 3-6s for completion. This delay breaks conversational flow.

Streaming LLMs send tokens as generated. Time-to-first-token becomes critical. Responses starting within 500-700ms feel natural even if completion takes longer.

Sentence-level streaming buffers tokens until complete sentences form before synthesis. This adds 200-500ms but produces natural speech without awkward cuts.

Smaller models like Claude Haiku or GPT-4o-mini provide 2-5x faster inference with 80-90% capability. For voice interactions, responsiveness often matters more than perfect reasoning.

Self-hosted models on GPUs achieve 100-300ms time-to-first-token. API-based models typically require 300-800ms. Local inference benefits latency-critical apps.

Speech Synthesis

Neural TTS produces human-like speech in 200-500ms. Concatenative synthesis offers 50-100ms but sounds robotic. Voice agents choose neural TTS despite latency cost.

Streaming TTS generates audio chunks as text arrives. First audio appears in 100-200ms. Real-time generation maintains 1:1 speed ratio between text and audio.

Audio buffering prevents choppy playback from network jitter. Too little buffer (50-100ms) causes gaps. Too much buffer (500ms+) increases latency. Adaptive buffering maintains 150-250ms optimally.

Network Optimization

Light speed limits create minimum latency. San Francisco to London requires 30ms each way, 60ms round-trip. Geographic latency cannot be eliminated, only minimized through edge deployment.

Edge computing places servers near users, reducing network RTT from 100ms to 10-20ms. This 80ms reduction significantly improves responsiveness.

WebRTC uses UDP transport to eliminate TCP head-of-line blocking and reduce jitter. However, ICE negotiation takes 1-3s initially. Persistent connections amortize this cost.

Optimization Strategies

Speculative processing starts work before certainty. During speech, partial ASR and predictive LLM inference save 200-500ms when correct. Wrong predictions waste compute but don't hurt UX.

Semantic caching matches similar queries to pre-computed responses. FAQ-style queries reduce from 2000ms to 50ms. Caching can cut median latency by 60-80%.

Parallel processing runs pipeline stages simultaneously. Start TTS on early LLM tokens while generation continues. Careful parallelization reduces end-to-end latency 30-50%.

Progressive enhancement starts with fast, lower-quality processing then upgrades asynchronously. Fast ASR provides immediate response, accurate ASR corrects in background if needed.

Measurement

Key metrics include time-to-first-token, time-to-first-audio, end-to-end latency, and component-level latencies. Track p50, p90, p99 percentiles, not just averages.

Real user monitoring captures actual performance across network conditions, geography, and devices. Lab testing misses real-world variability.

Future Directions

End-to-end voice models skip the ASR-LLM-TTS pipeline entirely, directly processing speech to speech. Early research shows 300-500ms end-to-end latency with 50-70% reduction potential.

Specialized voice LLMs optimized for conversation could provide 3-5x faster inference than general models. These trade broad capability for conversational excellence, targeting 50-150ms time-to-first-token.