CONFORMAL RISK CONTROL FOR CERTIFYING STREAMED GENERATION UNDER CONNECTIVITY LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

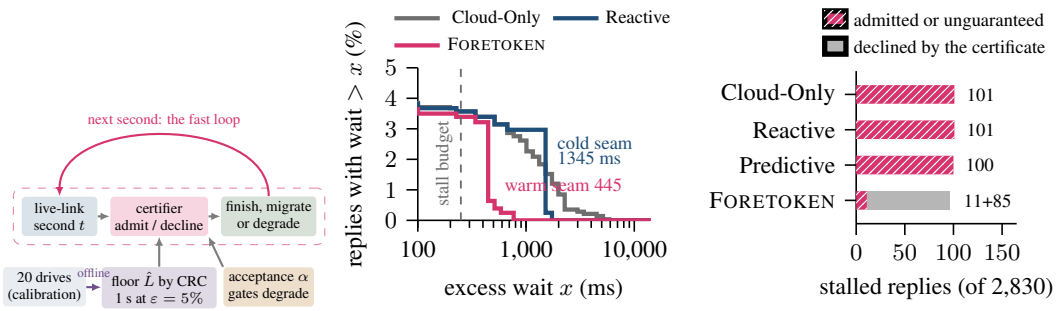
Conformal risk control certifies a threshold with a finite-sample guarantee, but only when the exchangeable unit is plentiful. We study it where the unit is scarce: a drive trace on a cellular network. A language-model reply streamed to a phone on a train is a sequential decision with a hidden hazard. The link can die before the last token arrives, and the serving stack learns of it last. Buffering cannot help, because future tokens do not exist yet. Reacting costs a loss-detection timeout plus a fallback start. We build FORETOKEN, a runtime that admits a response only under a completion certificate. The reply must either finish inside a certified floor on the link's remaining life, or hand off within that floor to an on-device drafter whose measured speculative-decoding acceptance rate clears a quality gate. The floor is a conformal-risk-control threshold over a grid of horizons with the trace as the exchangeable unit, so its realized per-second violation rate is bounded rather than assumed. On 21 public 5G traces (11.75 active hours, 2,830 replayed responses), FORETOKEN admits 96.7 % of launches with a response-level violation rate of 0.40 % against a 5% budget (trace-clustered 95% interval 0.18 to 0.69 %). It declines 92 launches, of which 92.4 % stall, and it holds p99 excess wait at 445 ms against 1,345 ms for a reactive hybrid. Three findings cut against our own premise. An RSRP-drop precursor covers 29.3 % of hard outages at a 19.5 % matched-control rate, and a learned ranker beats it at every false-alarm budget a phone would run. With 20 calibration traces the finite-sample term, not the hazard, decides what a 5% budget can certify: a one-second floor, eight seconds from 6%, and nothing at all below twenty traces. And when the budget sits at the hazard's own scale, calibrated-probability thresholds exceed it on up to 14 of 21 test traces while the conformal floor declines to certify. A completion guarantee does not need the outage to be predictable. It needs the hazard on the seconds it admits to be measured, bounded, and paid for.

1 INTRODUCTION

A commuter asks her assistant to summarize a contract, the reply streams at a comfortable reading pace, and the train enters the approach tunnel. The stream freezes, then resumes with three sentences missing. Nothing crashed. The serving stack did what it was built to do: it detected the loss, timed out, and failed over. It was built around the wrong question. It asks whether the link is up now, when the question a reply needs answered is whether the link will outlive the reply.

That second question looks like forecasting, and it partly is: cellular radio carries signals that sometimes degrade before service is lost. What breaks is not the forecast but what sits between the forecast and the reply. A threshold on a radio score fires early and often, or late and uselessly, and nothing downstream consumes its error rate. Buffering has nothing to buffer, since tokens are produced as they are sent. Reacting after the fact costs a detection timeout that must exceed normal jitter, plus the fallback's own start latency. The sum lands after the reader has watched the stream stop. The key tension is that the useful object is not a prediction of the outage but a guarantee about the reply. A guarantee has to be stated with its error budget and then audited, meaning its realized violation rate is measured against that budget. That makes this a calibration problem before

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) certifying loop (b) the wait tail, and where each system cuts it (c) who stalls, and whether a promise was made

Figure 1: **FORETOKEN at a glance.** **(a)** Every second, a reply is admitted only if finishing, migrating, or degrading fits inside a floor $\hat{L}$ on the link's remaining life. The floor is set offline by conformal risk control over whole traces, and the drafter's acceptance rate gates degrade. **(b)** Replay of 2,830 replies on 21 5G traces. About 3.5% of replies in every system wait past the 250 ms budget because they launch into an outage. Cloud-Only waits for the link, a cold drafter cuts the tail at 1345 ms, and FORETOKEN's resident drafter cuts it at 445 ms. **(c)** Of the FORETOKEN replies that stall, 85 had been declined and served without a promise. 11 of 2,738 admitted replies stall (0.40 %, trace-clustered interval in §5.3) against a 5% budget.

it is a systems problem. The natural tool is conformal risk control, and the awkward fact is that its exchangeable unit here is a whole trace, of which a corpus holds a few dozen.

FORETOKEN is that guarantee made operational (Figure 1a). Before a response is admitted, a certifier checks whether finishing, migrating to a second path, or degrading to an on-device drafter fits inside a floor $\hat{L}$ on the link's remaining life. The floor is not a point forecast. It is a threshold chosen by conformal risk control (Angelopoulos et al., 2024) over a grid of horizons, with the trace as the exchangeable draw. The expected fraction of admitted seconds followed by an interruption within $\hat{L}$ is therefore bounded by a stated budget ε on a fresh trace (Proposition 1). The drafter's quality is priced by a statistic the system already computes: the acceptance rate of its speculative drafts against the cloud verifier (Leviathan et al., 2023; Chen et al., 2023). We read that rate as a bound on how good offline generation will be, not as an accelerant.

We expected the precursor to carry the paper. It does not, and here is the surprising bit. On 21 public 5G traces, an RSRP drop of 6 dB over ten seconds fires before 29.3 % of the 58 hard outages, against a matched-control rate of 19.5 %. That is a real signal, 1.5 $\times$ chance, and far too weak to build a guarantee on. A gradient-boosted ranker on the same features does better at every false-alarm budget a phone would tolerate, the reverse of the small-n folklore, and it still leaves most outages unannounced. The certificate survives both results because it never needed the precursor. It needs the hazard on the seconds it admits to be small and measured, and on this corpus it is (Figure 1b,c). The second surprise is what the budget buys. With $n = 20$ calibration traces, the finite-sample correction $(1 - \varepsilon)/n$ consumes 0.25 of the 5 points a 5% budget offers, so the corrected condition certifies a floor of 1 s. From $\varepsilon = 6\%$ the same traces certify the full 8-second grid, and an eight-second floor at 5% needs 24 traces. We deploy at 5%, the budget of the registered claim, and show the step. The third surprise is what that price buys. On a harsher interruption definition, with ten calibration traces and budgets of 1 to 3%, calibrated-probability thresholds exceed their budget on up to 14 of 21 test traces. The conformal floor declines to certify there, because the finite-sample condition cannot be met, and it holds where it can. The same arithmetic faces any deployment of conformal risk control whose exchangeable unit is scarce: a clinic, a robot, a vehicle. Our curves say how the certifiable horizon grows with the count, and what breaks without it.

Key insight. A completion guarantee for a streamed response does not require the interruption to be predictable. It requires the hazard on the admitted seconds to be measured, bounded with its finite-sample price, and paid for by a fallback whose quality is measured too.

Contributions.

• **Certify admission with conformal risk control, and price the finite sample.** We state the certificate as a monotone loss over a horizon grid with the trace as the exchangeable unit, and we

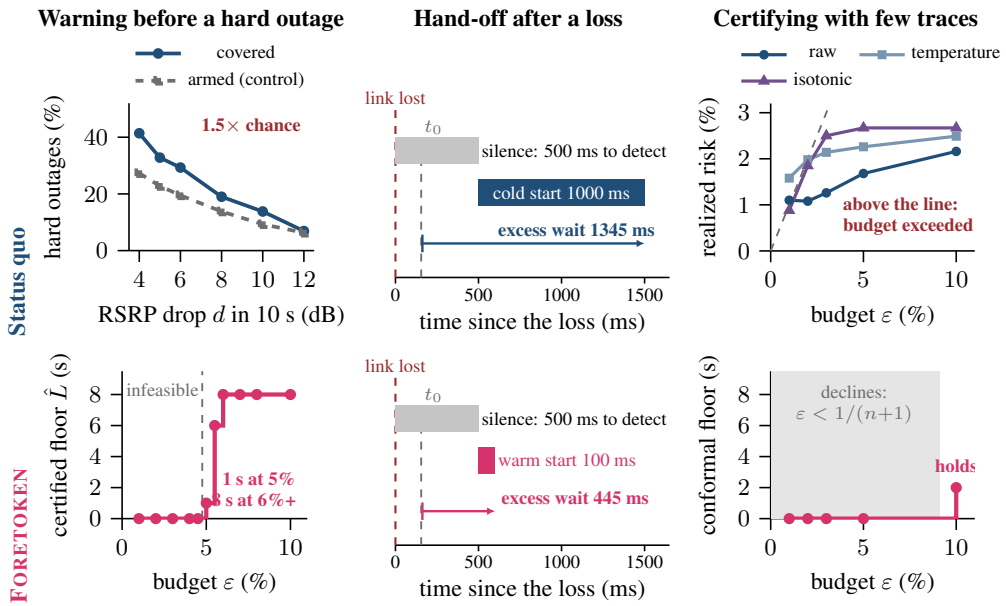


Figure 2: **What breaks, and what FORETOKEN changes.** Top row, the status quo. Bottom row, FORETOKEN. **Left:** an RSRP-drop rule fires before 29.3% of the 58 hard outages while armed on 19.5% of ordinary seconds. FORETOKEN instead certifies a floor on the link's remaining life by conformal risk control, and with $n = 20$ traces a 5% budget buys 1 s. **Middle:** after an unannounced loss the reader waits for detection plus the fallback's start, 1345 ms with a cold drafter and 445 ms with one kept resident by speculation. The degrade action is admitted only if the acceptance rate clears the gate. **Right:** under a harsher interruption definition (10 kbps) with ten calibration traces, calibrated-probability thresholds exceed budgets of 1 to 3%, while the conformal floor declines to certify below $1/(n+1)$ and holds at 10%.

audit it: 78 violations in 40,963 certified seconds (0.19 %) and 11 stalls in 2,738 admitted responses. We measure how the certifiable horizon grows with the number of traces (nothing at 5% below twenty, the full grid at 10% from twelve), and we show a setting where calibrated thresholds break the budget while the conformal floor declines (§3, §5.3, §5.4).

• **Measure how much warning exists.** We mine 588 micro-stalls, 58 hard outages, and 5 RAT losses from 11.75 active hours of public 5G traces. Only the RAT losses are reliably announced (4 of 5). Hard outages are announced 1.5 × chance at best, and a learned ranker beats a threshold rule at every false-alarm budget below 45 per hour (§2).

• **Price the fallback with a statistic that is free while online.** Speculative acceptance, ordinarily spent to go faster, becomes the quality bound that admits or refuses the degrade action. The gate opens as a step at $\alpha = q_{\min}$ and moves admission from 32.3 % to 96.7 % (§5.5).

• **Replay, ablate, and bound.** A trace-driven replay of 2,830 responses compares five systems and five ablations, sweeps every assumed constant, and replays migration against a second real trace under two correlation bounds (§5).

Scope. The forecast, the floor, and its audit are measured on real traces. The serving pipeline is a trace-driven replay with sourced or swept constants (Table 3), not a live deployment. We do not claim to predict interruptions. We claim to bound and audit the residual hazard on admitted seconds, for a client that paces display at reading speed.

2 How Much Warning Exists

Before building on a forecast we measure it (Hassan et al., 2022; Narayanan et al., 2020; Nicholson & Noble, 2008).

Data and definitions. We use the 21 continuous-demand download traces of the Irish 5G dataset (Raca et al., 2020): 16 driving and 5 static, sampled at 1 Hz with RSRP, RSRQ, SNR, CQI, cell identity, GPS, speed, and application throughput. An active second is one in which the client is downloading (42,303 of 43,208, 11.75 h). An interruption is a run of active seconds with zero downlink throughput[1]. A micro-stall lasts one or two seconds and a hard outage three or more. A RAT loss is a run of legacy-radio seconds (HSPA+, HSDPA, HSUPA, UMTS) after a 5G or LTE second. The corpus holds 588 micro-stalls (50.0 per active hour), 58 hard outages (4.9 per hour, median 4 s, longest 21 s, 46 in driving traces and 12 in static), and 5 RAT losses (median 531 s). The RAT losses are announced: 4 of 5 show a median RSRP slide of 9 dB in the ten seconds before the switch. They are not interruptions. Legacy seconds carry a median downlink of 2.5 Mbps, three orders of magnitude above a token stream's need, so we set them aside. The hard outages are what a reply has to survive.

A threshold rule. The natural precursor is a drop in RSRP over ten seconds (Hassan et al., 2022). Figure 2 (top left) sweeps the drop threshold d with the RSRP smoothed over two seconds. A rule "covers" an outage if it fires in the eight seconds before onset. It is "armed" on a second if it fired within the previous eight. Its matched-control rate is the fraction of ordinary seconds, outside any outage and its eight-second run-up, on which it is armed. The signal is real and small. Coverage runs from 41.4 % at $d = 4$ dB to 19.0 % at $d = 8$ dB against a control rate that tracks it, a lift of 1.1 to 1.5 $\times$ at every threshold, and micro-stalls sit on the control line. The deployed point, $d = 6$ dB, covers 17 of 58 outages with a median lead of 4 s at a 19.6 % duty. Two facts explain the ceiling. Static-trace outages have a flat radio (median RSRP range 1 dB before onset) and are application or backhaul stalls no radio rule can see: 0 of 12 covered, against 17 of 46 driving outages. And only 15.5 % of hard outages arrive within three seconds of a cell change, so the handover mechanism behind RAT-loss warnings (Hassan et al., 2022) governs a minority of zero-throughput events.

A learned ranker, and the reverse of the folklore. We expected a model trained on 51 positives to lose to the rule. It does not. Figure 3 evaluates a gradient-boosted classifier on the sixteen features of Appendix B, with 8:1 class weights and five folds grouped by trace so no trace crosses a fold boundary, at matched false-alarm budgets. At twelve alarms per hour it covers 18.3 % of held-out outages (three seeds: 15.7 to 21.6 %), where the best threshold rule at 12.2 alarms per hour covers 3.9 %. At forty-five per hour the ranker's 31.4 % still beats the rule's 15.7 %. Only near ninety alarms per hour, a duty no phone would run, do the two meet. For both forecasters the operating threshold is set on the evaluated seconds to hit the budget, so the curves compare score functions at matched duty rather than estimate a deployed alarm rate. The ranker learns something real from 51 events. Neither forecaster comes close to what a guarantee would need, and that is the finding. The certificate in §3 is built to hold when the forecaster is this weak.

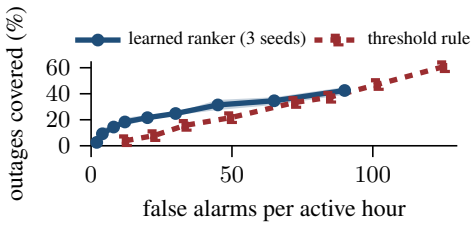


Figure 3: Coverage of held-out hard outages at a fixed false-alarm budget (8-second hold-off), trace-grouped folds. The ranker wins below 45 alarms per hour; the two meet near ninety.

3 FORETOKEN DESIGN

FORETOKEN is a thin runtime on the device plus a shim at the serving endpoint. It holds one guarantee: an admitted response does not stall past its budget except with a stated probability.

3.1 THE COMPLETION CERTIFICATE

Let W be the tokens remaining, r_c the verified cloud decode rate, t_0 the first-token latency on a healthy link (one round trip plus cloud prefill), $\hat{L}$ the certified floor of §3.2, m the measured cost of a recovery action, and α the drafter's current acceptance rate (§3.3). A response is covered iff at

[1]Zero is unambiguous. A 10 kbps threshold, the rate a 100 token/s stream needs with framing, raises the counts to 1,813 micro-stalls and 150 hard outages. §5.3 reports what that does to the certificate.

least one condition holds:

$$\text{(finish)} \quad t_0 + W/r_c \le \hat{L} \tag{1}$$

$$\text{(migrate)} \quad m_{\text{mig}} \le \hat{L} \ \wedge \ \text{an alternate path is up} \tag{2}$$

$$\text{(degrade)} \quad m_{\text{warm}} \le \hat{L} \ \wedge \ q(\alpha) \ge q_{\min} \tag{3}$$

where $q(\alpha)$ is the drafter's quality bound and $q_{\min}$ the application's floor. Admission requires coverage. A launch on a second whose floor is zero is declined and served without a guarantee. Each action is charged its measured price, not a worst case. With $\hat{L} = 1$ s a 60-token reply finishes inside the floor at $r_c = 100$ tokens/s. Longer replies are admitted on the degrade branch, whose validity rests on the drafter and the gate, not on the link.

3.2 The floor as a risk-controlling threshold

For each horizon h on a grid $\{1, \dots, 8\}$ s, define for a trace T

$$\ell_h(T) = \frac{\text{live active seconds of } T \text{ followed by a hard-outage onset within } h \text{ s}}{\text{live active seconds of } T}, \tag{4}$$

a fraction in $[0, 1]$ that is monotone non-decreasing in h by construction, because a longer claimed-safe horizon can only be violated by more onsets. That is the shape conformal risk control needs.

Proposition 1 (Trace-level risk control) *Let $T_1, \dots, T_n, T_{n+1}$ be exchangeable traces and ℓ_h as in Eq. 4. Let $\hat{L}$ be the largest h on the grid whose calibration mean $\frac{1}{n}\sum_{i \le n} \ell_h(T_i)$ is at most $\varepsilon - (1-\varepsilon)/n$, and $\hat{L} = 0$ if no h qualifies. Then $\mathbb{E}[\ell_{\hat{L}}(T_{n+1})] \le \varepsilon$.*

This is Theorem 1 of Angelopoulos et al. (2024) with $B = 1$ on a finite grid (the loss extended as a step function) and the trace, not the second, as the exchangeable draw. The proof is theirs. The guarantee is marginal over traces, not conditional on one. Three things matter here. First, the loss is defined on a fixed rule (a live second has nonzero downlink), so nothing is fitted on the calibration traces and every trace but the test one can calibrate. We run leave-one-trace-out with $n = 20$. Second, the guarantee is on the expected per-trace fraction over a fresh trace, and Figure 5 checks it trace by trace. Third, the condition is feasible only when $\varepsilon \ge 1/(n+1) = 4.76\%$. At $\varepsilon = 5\%$ it leaves an effective budget of $\varepsilon - (1-\varepsilon)/n = 0.25\%$. The calibration mean at $h = 1$ is 0.15 % and at $h = 2$ is 0.26 %, so the certified floor is 1 s, realized at 0.19 % against the 0.25 % bound. At $\varepsilon = 5.5\%$ the floor is 6 s, and from 6% it is 8 s, the top of the grid (Figure 2, bottom left, and Figure 12 for the calibration loss by horizon). The calibration mean at $h = 8$ is 0.98 %, so an eight-second floor at 5% needs $n \ge (1-\varepsilon)/(\varepsilon - \ell_8)$, that is 24 traces. Figure 4 makes the price empirical by subsampling the calibration traces. A 5% budget certifies nothing below $n = 20$. A 6% budget climbs from almost nothing at $n = 16$ to the full grid at $n = 20$. A 10% budget certifies the full grid from $n = 12$. What breaks at small n is not validity, which Proposition 1 keeps, but the horizon. The guarantee stays honest by certifying less, and the count of exchangeable units, not the hazard, decides how much.

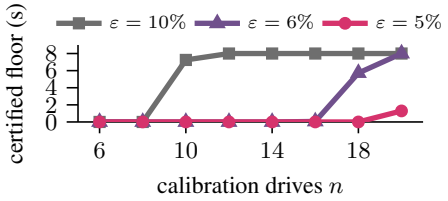


Figure 4: Certified floor against the number of calibration traces (50 subsamples per leave-one-out fold). The count of exchangeable units, not the hazard, sets the horizon.

Drift is handled by monitoring, not by online-valid machinery such as adaptive conformal inference (Gibbs & Candès, 2021). When realized hazard exceeds budget the floor collapses toward zero, and FORETOKEN degrades into the reactive baseline rather than into overconfidence.

3.3 Acceptance as a runtime verifier

The degrade action needs a quality bound, and buying one with extra traffic would defeat the point. While the link is up, the device drafts tokens with a small model M_d and the cloud verifier accepts or rejects them (Leviathan et al., 2023; Chen et al., 2023). The acceptance rate α is therefore a continuously updated measurement of how often the small model already says what the large one would. We set $q(\alpha) = \alpha$, the fraction of fallback tokens the verifier would have kept (q may be any

monotone map), tracked per context class with a lower confidence bound. Cascades and judge-style decoders reuse the same statistic to decide when the small model may answer alone (Narasimhan et al., 2025; Bachmann et al., 2025). FORETOKEN reuses it to decide, before the link dies, whether offline generation is good enough to promise. Speculation has a second role. It keeps M_d resident with the prefix prefilled, so a hand-off costs a warm start ($m_{\text{warm}} = 100$ ms) rather than a cold one ($1,000$ ms).

3.4 ACTIONS

Finish (Eq. 1) needs only arithmetic. Degrade hands the remainder to M_d with the prefix intact, and the pacer spends the reader's headroom to hide the seam. Migrate moves the decode session to a second path. The expensive part is context, so the endpoint keeps state warm and streams deltas, as CacheGen does for KV state (Liu et al., 2024). Slack inside the floor can be spent on other dials. Appendix C evaluates one, routing the verifier to a lower-carbon region when the finish condition still holds with the detour added.

4 EXPERIMENTAL SETUP

We separate what is measured from what is replayed. The interruptions, precursors, floor, and audit are measured on the traces. The serving pipeline is a trace-driven replay. A response launches every 15 active seconds at a uniformly random sub-second offset, lengths cycle through 60, 120, and 240 tokens, and the simulation advances in 100 ms steps over the trace's own per-second link state (2,830 responses). Tokens arrive from the cloud after $t_0 = 155$ ms (a 75 ms round trip, the median of the traces' 3,208 ping samples, plus 80 ms prefill). They arrive at $r_c = 100$ tokens/s while the downlink is nonzero, pause while it is zero, and resume after one round trip. The client paces display at 4.8 tokens/s[2] from a buffer, so the metric is honest about what a reader sees. *Excess wait* is first-token delay beyond t_0 plus any later interval in which the display has nothing to show. A response has *stalled* when its excess wait exceeds 250 ms. The drafter decodes at 24.65 tokens/s (TinyLlama-1.1B, q4_k, iPhone 14 Pro, single stream, per MELTing Point (Laskaridis et al., 2024); three other classes swept). Loss detection costs 500 ms of silence, above cellular jitter. An unannounced loss handed to a warm drafter therefore costs 445 ms of excess wait, and to a cold one 1,345 ms. §5.5 sweeps both. Quality is the fraction of tokens the verifier produced plus α times the fraction the drafter produced alone, with $\alpha = 0.60$ and $q_{\min} = 0.45$. Baselines: Cloud-Only and Device-Only bound the space. A Reactive hybrid hands off to a cold drafter after observed loss. Predictive adds the arming rule of §2 so a hand-off while armed is warm. FORETOKEN adds speculation (warm drafter always), the certificate, and the gate. Without a pacer every mid-stream outage becomes visible and the certificate protects less. We say so rather than model it.

5 RESULTS

Five questions, in order. What changes end to end. Where the change comes from. Whether the certificate is honest. Why risk control rather than calibration. And how the answers move with the constants we assumed.

5.1 END TO END

Table 1 shows where each mechanism pays. Cloud-Only is cheap in the mean and brutal in the tail (p99 1,871 ms). Reactive trims that tail to 1,345 ms, the cold seam. Predictive's arming rule shaves the mean but not the tail, which is made of unannounced losses. FORETOKEN holds p99 at 445 ms, the warm seam, a 67 % cut against Reactive, at the lowest stall rate, 3.39 %. That tail gain is the warm drafter speculation keeps resident, not the certificate (§5.2). What the certificate adds is visible only by splitting the population. Of 2,830 launches, FORETOKEN admits 2,738 (96.7 %) and declines the 92 that arrive on a dead second. Admitted replies wait 2.7 ms on average, at least 99% of them wait nothing beyond t_0, and 11 of them stall (0.40 %). Declined replies wait 418.3 ms and

[2] About 220 words per minute at 1.3 tokens per word, inside the silent-reading range of Brysbaert (2019); swept from 3.3 to 6.5.

Table 1: End-to-end replay over 2,830 responses on 21 traces at the nominal constants of §4. Excess wait, p99, and TTFT in ms. Quality as defined in §4. Stall is the fraction of all responses past the 250 ms budget. Bold marks the best value in each column.

System	Wait	p99	TTFT	Quality	Stall (%)
Cloud-Only	54.9	1,871	207	**1.000**	3.57
Device-Only	845.0	845	1,000	0.600	100.00
Reactive	44.1	1,345	198	0.987	3.57
Predictive	37.5	1,345	192	0.987	3.53
FORETOKEN	**16.2**	**445**	**171**	0.987	**3.39**

Table 2: Ablations of FORETOKEN at the nominal constants. "Admit" is the fraction of launches certified; "Viol." the stalls among admitted replies as a count and a rate against the 5% budget; the no-certificate row admits every launch and so has neither. Units as in Table 1.

Variant	Wait	p99	Quality	Admit (%)	Viol.	Rate (%)
FORETOKEN (nominal)	16.2	445	0.987	96.7	11 /2,738	0.40
no speculation	44.1	1,345	0.987	32.3	5 /915	0.55
no certificate	16.2	445	0.987	all	n/a	n/a
+ precursor in floor	16.2	445	0.987	78.1	8 /2,209	0.36
+ pre-emptive hand-off	16.3	445	0.983	78.1	8 /2,209	0.36

stall 85 times (92.4 %). The floor separates the two populations, and the guarantee is a statement about the first.

5.2 WHERE THE CHANGE COMES FROM

Table 2 isolates each mechanism, and two of its rows are the negative results we promised. Removing speculation returns the drafter to a cold start. The degrade branch of Eq. 3 no longer fits inside a 1 s floor, admission falls to 32.3 % (the 60-token replies that finish), and p99 returns to 1,345 ms. Removing the certificate keeps the warm drafter and therefore the tail, at 445 ms, while giving up the split of the previous paragraph. Then the two rows we expected to help. Folding the threshold rule of §2 into the floor, so that armed seconds are not certified, removes admissions (78.1 % against 96.7 %) and leaves the violation rate where it was (0.36 % against 0.40 %). A 1.5 × lift is too small to move the certified hazard, and its duty is paid in declined replies. Adding a pre-emptive hand-off on arming, the mechanism a forecast is supposed to enable, fires 57 times, lowers quality from 0.987 to 0.983, and changes nothing else. With a paced display the buffer covers a mid-stream loss, and the exposure that remains sits at the start of the reply. We report both rows and deploy neither (Figure 14 sweeps both together). The same null is why no ranker-based Predictive is run: no mid-stream action reaches that interval. Clustered intervals of these rows overlap the nominal one. The gate's own ablation is the sweep of Figure 7.

5.3 IS THE CERTIFICATE HONEST

We audit the guarantee at both units. At the unit of Proposition 1, across 40,963 certified seconds on 21 traces, 78 are followed by a hard outage within the 1 s floor. That is a realized rate of 0.19 % against the 5% budget, under the corrected bound of 0.25 %. Seconds are correlated within a trace, so we report a trace-clustered bootstrap interval, 0.12 to 0.28 %, 2.0 × the naive per-second width. At the unit a user cares about, 11 of 2,738 admitted replies stall (0.40 %, clustered interval 0.18 to 0.69 %). The response-level number is an audit, not a theorem. Proposition 1 bounds the per-second loss, and a 240-token reply admitted on the degrade branch is exposed for 2.55 s, relying on the buffer and the warm drafter for what the floor does not cover. Figure 5 checks the per-second rate trace by trace. The worst trace reaches 0.82 % on 364 certified seconds, the median is 0.12 %, and no trace exceeds the budget. The unweighted mean of the per-trace rates, the quantity Proposition 1 bounds, is 0.22 %. At $\varepsilon = 10\%$

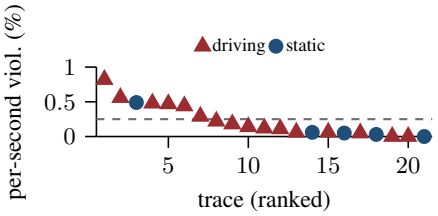


Figure 5: Per-second violation rate of the 1 s floor on each of the 21 traces, ranked; dashed: the corrected bound (0.25 %). The 5% budget is off the chart.

the same procedure certifies 8 s on every fold and realizes 0.88 % (0.61 to 1.21 %), a budget it never approaches. The tighter budget binds. The interruption definition matters too. At a 10 kbps threshold the corrected 5% condition admits no horizon at $n = 20$, and 10% certifies 8 s at 2.67 %.

5.4 WHY RISK CONTROL RATHER THAN CALIBRATION

Why not calibrate a hazard classifier's probabilities and threshold them at the budget? We answer on a nested split that gives every method the same data. For each test trace, 10 traces train a gradient-boosted classifier per horizon and 10 calibrate it, by Platt scaling (Platt, 1999), temperature scaling (Guo et al., 2017), isotonic regression (Zadrozny & Elkan, 2002), or histogram binning (Zadrozny & Elkan, 2001). A second's floor is the longest horizon whose calibrated probability is at most ε. The conformal floor uses the same 10 calibration traces over the classifier's declared-safe seconds. Figure 6 sweeps ε (realized risks in Table 4). On validity the calibrated methods do not separate. Every one realizes about 0.88 % at every budget, because this corpus's hazard sits far below any budget a deployment would set, and every one certifies close to the full grid. The conformal floor certifies nothing below $\varepsilon = 9.1\%$, the feasibility line at $n = 10$, and at 10% it certifies 7.08 s. On this corpus, then, the calibrated thresholds are valid because the hazard is low, and nothing certifies that they would be on a harsher one. The conformal threshold refuses to promise what ten traces cannot support.

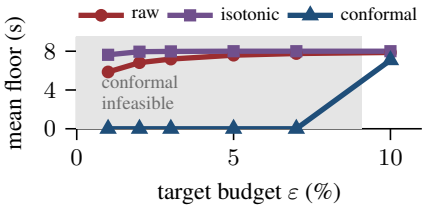


Figure 6: Mean certified floor of calibrated-probability thresholds and of the conformal floor on identical nested folds (10 train, 10 calibrate, one test trace). The shaded band is where the conformal condition is infeasible at $n = 10$, and the floor is zero there by construction. Realized risks are in Table 4 and Figure 10.

When the budget binds. To see the difference we make the hazard larger and the budget tighter (Figure 2, right, and Table 5). Under a 10 kbps interruption definition the corpus holds 150 hard outages, and the per-second hazard at eight seconds rises to 2.99 %. On the same nested folds, at budgets of 1 to 3%, every calibrated threshold exceeds its budget on some test traces, temperature scaling on 14 of 21 at 1% and Platt scaling on 10 of 21 at 2%, and 2 of the fifteen method-budget pairs exceed the budget in the pooled rate as well. The conformal floor certifies nothing at those budgets, because $\varepsilon < 1/(n+1) = 9.1\%$ at $n = 10$, and at 10% it certifies 2.00 s at a realized 0.71 % against its corrected target of 1%. Restricting to the 12 highest-hazard traces, with 5 calibration traces, sharpens both halves: temperature scaling exceeds a 1% budget on 11 of 12 test traces, and the conformal floor declines at every budget up to 10%, because nothing is feasible below 16.7 %. Calibration buys horizon and can break the budget. Risk control buys a guarantee with its price attached, and says so when the price cannot be paid.

5.5 SENSITIVITY, WITH SOURCED RANGES

Every constant in §4 names its source or is marked an assumption, and every point below is a harness run (Table 3 lists all of them). Three sweeps change the picture. The rest do not.

The gate is a step. At $\alpha = 0.30$, below the relaxed $q_{\min} = 0.45$, the degrade branch closes and admission falls from 96.7 % to 32.3 %, the finish-only set. It reopens at 96.7 % the instant α clears $q_{\min}$, and a strict gate at $q_{\min} = 0.60$ opens one step later (Figure 7). Higher acceptance then buys quality, from 0.978 to 0.995, and nothing else.

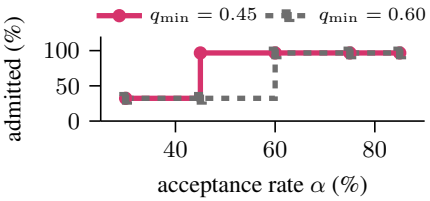


Figure 7: Admission against the drafter's acceptance rate under two quality gates at the 5% budget; each curve is a step at $q_{\min}$ because the degrade branch is a binary condition.

The seam is the detection timeout. The tail of every hybrid is detect + $m - t_0$. A 100 ms timeout drops FORETOKEN's p99 from 445 to 45 ms and its stall rate to 0.00 %. A one-second timeout raises p99 to 1,045 ms. Warm-start cost moves the same tail linearly (p99 395 to 1,145 ms over 50 to 800 ms) and the violation rate not at all.

A drafter slower than the reader starves the reader. Drafter class has no effect at 4.8 tokens/s, because every class in the MELTing Point sweep decodes faster than a person reads. Raise the reader to 6.5 tokens/s and the 7B model at 5.99 tokens/s cannot keep up. Replies handed to it wait 2,452 ms on average, against 443 ms at 4.8 tokens/s, and every one of them stalls. Eq. 3 should carry $r_d \geq r_{\mathrm{read}}$ where drafters are slow.

What does not move. Reading rate from 3.3 to 6.5 tokens/s and round-trip time from 67 to 232 ms leave the violation rate under 0.80 %. Cloud decode rate from 30 to 200 tokens/s leaves p99 at 445 ms and changes only which replies finish inside the floor (Figure 13, Appendix D).

5.6 Migration against a second real trace

The single-operator traces carry no second path, so we pair each response with another trace as a stand-in alternate under two bounds. Under the independent bound the alternate is a different trace at a random offset. Under the shared-outage bound it is forced down with probability p_{share} whenever the home path is. Migration is admitted as Eq. 2 states, with a switch cost of one round trip plus 200 ms of context transfer. On the 154 replies that needed a hand-off it is a quality mechanism, not a latency one. Under the independent bound it lifts the subset's quality from 0.768 to 0.981 because tokens stay verified, and it raises the subset's wait from 297.9 to 433.7 ms, since a path switch costs more than a warm drafter's seam (Figure 11). The benefit decays monotonically with the assumed correlation, to a quality of 0.794 at $p_{\mathrm{share}} = 0.9$. Table 1 does not move.

6 Related Work

Conformal prediction and risk control. Split conformal prediction gives distribution-free coverage for a black-box predictor (Angelopoulos & Bates, 2023). Risk-controlling prediction sets bound a set-valued loss (Bates et al., 2021), Learn then Test handles the multiple-testing version (Angelopoulos et al., 2025), and conformal risk control gives the monotone-loss form that Proposition 1 instantiates (Angelopoulos et al., 2024). Conformal Language Modeling points that machinery at a sampling stopping rule (Quach et al., 2024) and CALM at early exit (Schuster et al., 2022). We point it at a completion guarantee whose loss is a user-visible stall and whose exchangeable unit is a trace. Adaptive conformal inference tracks shift online (Gibbs & Candès, 2021), a stronger claim than ours.

Cellular measurement and prediction. The lineage runs from commute-trace video adaptation (Riiser et al., 2013) and connectivity forecasting (Nicholson & Noble, 2008; Schulman et al., 2010; Higgins et al., 2012) to the datasets we mine (Raca et al., 2018; 2020), mapped throughput (Narayanan et al., 2020), on-phone analytics (Li et al., 2016), and the handover dissection we tested (Hassan et al., 2022). Each feeds a prediction to something that can wait.

Speculative decoding and serving. Speculative decoding began as verification-preserving acceleration (Leviathan et al., 2023; Chen et al., 2023) and reached phones and edge clusters (Xu et al., 2025; Miao et al., 2024; Narayan et al., 2025). Cascades and judge decoding reuse the acceptance signal as a decision statistic (Narasimhan et al., 2025; Bachmann et al., 2025). vLLM and its successors define the serving fabric (Kwon et al., 2023; Patel et al., 2024; Stojkovic et al., 2025), and CacheGen streams the KV state that migration reuses (Liu et al., 2024).

7 Limitations and Conclusion

The traces are 2019 to 2020 vintage, one Irish operator, NSA-era 5G, driving and static only, and benign (longest hard outage 21 s). The floor's value here is therefore a bound and an accounting, not evidence against a long tunnel. The trace is the honest exchangeable unit and also why $n = 20$ limits the horizon. Exchangeability across traces from one operator in one region is an assumption, and the response-level rate is audited, not proven. The replay assumes a paced display, a 500 ms detection timeout, and a resident drafter, and it reports no live acceptance rate, memory footprint, or battery draw. What we can say is narrower than the claim we set out to make and, we think, more useful. A completion certificate does not need the loss announced. It needs the hazard on admitted seconds measured, bounded with its finite-sample price, and paid for by a measured fallback. On 21 traces that bound holds at every unit we audited.

AI USE STATEMENT

We used a large language model as an assistant for drafting and editing the text, for drafting the TikZ and PGFPlots figure code, and for drafting and refactoring the analysis pipeline (the event miner, the conformal floor and its audit, the trace-driven replay harness, the sweeps, and the script that writes every number and figure coordinate into the paper) from our own specifications and the public traces. The research question, the design of the completion certificate and the floor, the choice of data and of the interruption definitions, the experimental settings, and the interpretation of the results are the authors' own. Every reported number is recomputed by the released scripts from the public dataset, and the authors reviewed all AI-assisted text and code. We take full responsibility for the content of this paper.

ETHICS STATEMENT

This work raises no human-subject concerns. The traces are a public dataset released by its authors (Raca et al., 2020). The design keeps location-bearing radio features on the device by construction and exports only an admit-or-decline decision. The traces carry the collectors' GPS tracks, which we do not republish. The optional carbon-routing dial of Appendix C uses public grid-intensity data (Ember, 2024; RTE, 2024). Routing a verifier request to another region moves user text across a jurisdiction, so a deployment must apply its data-residency rules before enabling that dial. A certificate that admits replies could be misread as a promise that the link will hold. We state its unit and its budget wherever it is quoted.

REPRODUCIBILITY STATEMENT

Every number in this paper, including every figure coordinate and every macro in the text, is written by one script in the released artifact from the public Irish 5G dataset. The script runs the miner of §2, the floor and audit of §3.2 and §5.3, the nested baseline comparison of §5.4, and the replay harness and sweeps of §5. Appendix A specifies the harness, Table 3 lists every constant with its source or its sweep, and Appendix B lists the ranker's features. The artifact is provided as anonymized supplementary material and runs on a laptop CPU in under two minutes.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: https://doi.org/10.1561/2200000101.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2), 2025. doi: https://doi.org/10.1214/24-AOAS1998.

Gregor Bachmann, Sotiris Anagnostidis, Albert Pumarola, Markos Georgopoulos, Artsiom Sanakoyeu, Yuming Du, Edgar Schönfeld, Ali Thabet, and Jonas Kohler. Judge decoding: Faster speculative sampling requires going beyond model alignment. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, 2025. URL https://openreview.net/forum?id=mtSSFiqW6y.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):43:1–43:34, 2021. doi: https://doi.org/10.1145/3478535.

Marc Brysbaert. How many words do we read per minute? A review and meta-analysis of reading rate. *Journal of Memory and Language*, 109:104047, 2019. doi: https://doi.org/10.1016/j.jml.2019.104047.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

Andrew A. Chien, Liuzixuan Lin, Hai Nguyen, Varsha Rao, Tristan Sharma, and Rajini Wijayawardana. Reducing the carbon impact of generative AI inference (today and in 2035). In *Proceedings of the 2nd Workshop on Sustainable Computer Systems (HotCarbon '23)*. ACM, 2023. doi: https://doi.org/10.1145/3604930.3605705.

Ember. Yearly electricity data. https://ember-energy.org/data/yearly-electricity-data/, 2024. Data year 2023; lifecycle emissions intensity by country. Accessed 2026-09-22.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, volume 34, pp. 1660–1672. Curran Associates, Inc., 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *PMLR*, pp. 1321–1330, 2017.

Ahmad Hassan, Arvind Narayanan, Anlan Zhang, Wei Ye, Ruiyang Zhu, Shuowei Jin, Jason Carpenter, Z. Morley Mao, Feng Qian, and Zhi-Li Zhang. Vivisecting mobility management in 5G cellular networks. In *Proceedings of the ACM SIGCOMM 2022 Conference*, pp. 86–100. ACM, 2022. doi: https://doi.org/10.1145/3544216.3544217.

Brett D. Higgins, Jason Flinn, T. J. Giuli, Brian Noble, Christopher Peplin, and David Watson. Informed mobile prefetching. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, pp. 155–168. ACM, 2012. doi: https://doi.org/10.1145/2307636.2307651.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP '23)*, pp. 611–626. ACM, 2023. doi: https://doi.org/10.1145/3600006.3613165.

Stefanos Laskaridis, Kleomenis Katevas, Lorenzo Minto, and Hamed Haddadi. MELTing Point: Mobile evaluation of language transformers. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom '24)*, pp. 890–907. ACM, 2024. doi: https://doi.org/10.1145/3636534.3690668.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning (ICML '23)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Baolin Li, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. Clover: Toward sustainable AI with carbon-aware machine learning inference service. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '23)*. ACM, 2023. doi: https://doi.org/10.1145/3581784.3607034.

Baolin Li, Yankai Jiang, Vijay Gadepally, and Devesh Tiwari. Sprout: Green generative AI with carbon-efficient LLM inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP '24)*, pp. 21799–21813. Association for Computational Linguistics, 2024. doi: https://doi.org/10.18653/v1/2024.emnlp-main.1215.

Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng, and Tao Wang. MobileInsight: Extracting and analyzing cellular network information on smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*, pp. 202–215. ACM, 2016. doi: https://doi.org/10.1145/2973750.2973751.

Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, Michael Maire, Henry Hoffmann, Ari Holtzman, and Junchen Jiang. CacheGen: KV cache compression and streaming for fast large language model serving. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pp. 38–56. ACM, 2024. doi: https://doi.org/10.1145/3651890.3672274.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. SpecInfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS '24)*, pp. 932–949. ACM, 2024. doi: https://doi.org/10.1145/3620666.3651335.

Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Seungyeon Kim, Neha Gupta, Aditya Krishna Menon, and Sanjiv Kumar. Faster cascades via speculative decoding. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, 2025. URL https://openreview.net/forum?id=vo9t20wsmd.

Avanika Narayan, Dan Biderman, Sabri Eyuboglu, Avner May, Scott Linderman, James Zou, and Christopher Ré. Cost-efficient collaboration between on-device and cloud language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 45682–45719. PMLR, 2025. URL https://proceedings.mlr.press/v267/narayan25a.html.

Arvind Narayanan, Eman Ramadan, Rishabh Mehta, Xinyue Hu, Qingxu Liu, Rostand A. K. Fezeu, Udhaya Kumar Dayalan, Saurabh Verma, Peiqi Ji, Tao Li, Feng Qian, and Zhi-Li Zhang. Lumos5G: Mapping and predicting commercial mmWave 5G throughput. In *Proceedings of the ACM Internet Measurement Conference (IMC '20)*, pp. 176–193. ACM, 2020. doi: https://doi.org/10.1145/3419394.3423629.

Anthony J. Nicholson and Brian D. Noble. BreadCrumbs: Forecasting mobile connectivity. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom '08)*, pp. 46–57. ACM, 2008. doi: https://doi.org/10.1145/1409944.1409952.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, and Ricardo Bianchini. Splitwise: Efficient generative LLM inference using phase splitting. In *Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA '24)*, pp. 118–132. IEEE, 2024. doi: https://doi.org/10.1109/ISCA59077.2024.00019.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi Jaakkola, and Regina Barzilay. Conformal language modeling. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Darijo Raca, Jason J. Quinlan, Ahmed H. Zahran, and Cormac J. Sreenan. Beyond throughput: A 4G LTE dataset with channel and context metrics. In *Proceedings of the 9th ACM Multimedia Systems Conference (MMSys '18)*, pp. 460–465. ACM, 2018. doi: https://doi.org/10.1145/3204949.3208123.

Darijo Raca, Dylan Leahy, Cormac J. Sreenan, and Jason J. Quinlan. Beyond throughput, the next generation: A 5G dataset with channel and context metrics. In *Proceedings of the 11th ACM Multimedia Systems Conference (MMSys '20)*, pp. 303–308. ACM, 2020. doi: https://doi.org/10.1145/3339825.3394938.

Ana Radovanović, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, Saurav Talukdar, Eric Mullen, Kendal Smith, MariEllen Cottman, and Walfredo Cirne. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems*, 38(2):1270–1280, 2023. doi: https://doi.org/10.1109/TPWRS.2022.3173250.

Haakon Riiser, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen. Commute path bandwidth traces from 3G networks: Analysis and applications. In *Proceedings of the 4th ACM Multimedia Systems Conference (MMSys '13)*, pp. 114–118. ACM, 2013. doi: https://doi.org/10.1145/2483977.2483991.

RTE. 2023 annual electricity review: Key findings. https://analysesetdonnees.rte-france.com/en/annual-review-2023/keyfindings, 2024. Accessed 2026-09-22.

Aaron Schulman, Vishnu Navda, Ramachandran Ramjee, Neil Spring, Pralhad Deshpande, Calvin Grunewald, Kamal Jain, and Venkata N. Padmanabhan. Bartendr: A practical approach to energy-aware cellular data scheduling. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom '10)*, pp. 85–96. ACM, 2010. doi: https://doi.org/10.1145/1859995.1860006.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022.

Jovan Stojkovic, Chaojie Zhang, Íñigo Goiri, Josep Torrellas, and Esha Choukse. DynamoLLM: Designing LLM inference clusters for performance and energy efficiency. In *2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA '25)*, pp. 1348–1362. IEEE, 2025. doi: https://doi.org/10.1109/HPCA61900.2025.00102.

Thanathorn Sukprasert, Abel Souza, Noman Bashir, David Irwin, and Prashant Shenoy. On the limitations of carbon-aware temporal and spatial workload shifting in the cloud. In *Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys '24)*, pp. 924–941. ACM, 2024. doi: https://doi.org/10.1145/3627703.3650079.

Daliang Xu, Wangsong Yin, Hao Zhang, Xin Jin, Ying Zhang, Shiyun Wei, Mengwei Xu, and Xuanzhe Liu. EdgeLLM: Fast on-device LLM inference with speculative decoding. *IEEE Transactions on Mobile Computing*, 24(4):3256–3273, 2025. doi: https://doi.org/10.1109/TMC.2024.3513457.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, pp. 609–616, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pp. 694–699. ACM, 2002. doi: https://doi.org/10.1145/775047.775151.

A Replay Harness Specification

The timeline is each trace's active seconds in order at 1 Hz, and the simulation step is 100 ms. A response with W tokens launches at active second t_0 plus a uniformly random sub-second offset. The cloud request leaves at launch and waits while the downlink is zero. The first token lands one round trip plus prefill later. Later tokens land at r_c tokens/s while the downlink is nonzero, pause while it is zero, and resume after one round trip. The display consumes tokens from a buffer at r_{read} tokens/s from the moment the first token lands. The reader is starved when the buffer is empty and the reply is incomplete. Excess wait is first-token time beyond t_0 plus total starvation. A reply stalls when excess wait exceeds the budget. A hand-off to the drafter occurs when the link has been silent for the detection timeout and the buffer cannot cover the drafter's start latency. The drafter then produces the remainder at r_d tokens/s after that latency. For FORETOKEN the drafter is warm, because speculation keeps it resident. For Reactive it is cold. For Predictive it is warm while the arming rule is armed. Quality is the verified-token fraction plus α times the drafter-only fraction. Carbon per reply is verified tokens times the region's intensity plus drafter tokens times the device's energy ratio times the home intensity. The detour intensity is Ember's lifecycle figure for France, 29 $\mathrm{MtCO_2e}$ over RTE's 494.7 TWh of 2023 generation. RTE's own direct-emission figure,

32 g/kWh, would enlarge the cut. The link model has no transport dynamics. A real stream recovers through retransmission and congestion control after an outage, so the harness is optimistic for every system equally. Carbon counts decode tokens, not the verifier's rejection overhead. Migration, when enabled, is tried before degrade at any hand-off. If the alternate trace is up (and not forced down by the shared-outage draw) and the switch seam fits in the buffer, the remainder streams over the alternate at full quality. The certifier's per-update arithmetic costs 0.08 μs on a laptop core, so its cost is not a factor. Launch offsets are deterministic hashes of the trace name and launch time, the ranker uses seeds 0, 1, and 2, and the clustered bootstrap draws 5,000 trace resamples from a fixed seed. A re-run therefore reproduces every number exactly. The whole pipeline, mining through sweeps, runs in under two minutes on one core.

Table 3: Every constant in the replay, its nominal value, its source, and the range swept. "Assumption" marks values with no measurement behind them.

Constant	Nominal	Source	Swept
Reading rate	4.8 tok/s	Brysbaert (2019) at 1.3 tok/word	3.3 to 6.5
Cloud decode rate r_c	100 tok/s	assumption (single stream)	30 to 200
Drafter rate r_d	24.65 tok/s	Laskaridis et al. (2024), iPhone 14 Pro	5.99 to 24.65
Round trip	75 ms	median of 3,208 trace pings	67 to 232
Cloud prefill	80 ms	assumption	fixed
Cold drafter start	1,000 ms	assumption (load + prefill)	fixed
Warm drafter start	100 ms	assumption	50 to 800
Detection timeout	500 ms	assumption (above jitter)	100 to 1000
Stall budget	250 ms	assumption	fixed
Launch spacing	15 s	assumption	fixed
Response lengths	60/120/240	assumption	up to 500/1000/2000
Acceptance α	0.60	assumption	0.30 to 0.85
Gate $q_{\min}$	0.45	assumption	0.45, 0.60
Grid intensity, home (App. C)	283 g/kWh	Ireland 2023 (Ember, 2024)	fixed
Grid intensity, detour (App. C)	59 g/kWh	France 2023 (Ember, 2024; RTE, 2024)	fixed
Carbon detour (App. C)	30 ms	assumption	10 to 80
Device/cloud energy per token (App. C)	0.25	assumption	0.1 to 0.5
Context transfer (migrate)	200 ms	assumption	100 to 500
Budget ε	5%	registered claim	5 to 10

B Ranker Features and Event Geometry

The sixteen features, all computable on the phone at 1 Hz: RSRP, RSRQ, SNR, CQI, RSSI, neighbor RSRP, neighbor RSRQ, speed; RSRP change over 5 and 10 s; RSRQ, SNR, and CQI change over 10 s; minimum and standard deviation of RSRP over 10 s; cell changes in the last 10 s. Missing radio values are carried forward. Figure 8 places every mined event by the lead time the deployed rule gives it and its duration.

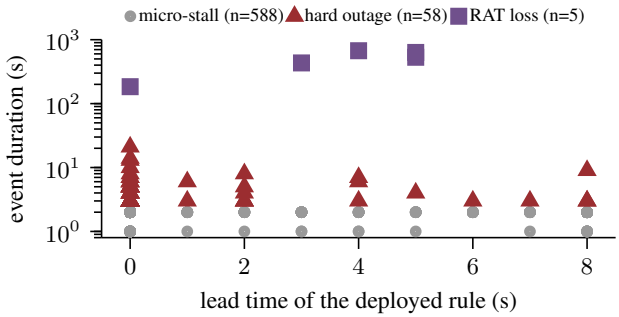


Figure 8: Every mined event by the lead time the 6 dB rule gives it (0 means it fired nowhere in the eight seconds before onset) and its duration on a log scale. RAT losses are long and announced; hard outages are short and mostly not; micro-stalls sit at chance.

C An Optional Dial: Spending Certified Slack on Carbon

The finish condition of Eq. 1 leaves slack whenever a reply finishes well inside the floor. One way to spend it is on where the verifier runs. If the finish condition still holds with a greener region's extra round trip added to t_0, the verifier request is routed there. This is the one place FORETOKEN

spends rather than saves, and it is evaluated here rather than in the body because it is an application of the certificate, not part of it.

We report carbon relative to Cloud-Only, with grid intensities of 283 and 59 gCO_2e/kWh for the home and detour regions (Ireland and France in 2023 (Ember, 2024; RTE, 2024)), a 30 ms detour, and a device-to-cloud energy ratio per token of 0.25. Only the ratios enter the result. The detour intensity is Ember's lifecycle figure for France, 29 $MtCO_2e$ over RTE's 494.7 TWh of 2023 generation. RTE's own direct-emission figure, 32 g/kWh, would enlarge the cut. Carbon counts decode tokens, not the verifier's rejection overhead. Carbon-aware scheduling supplies the routing mechanism (Radovanović et al., 2023; Sukprasert et al., 2024; Chien et al., 2023; Li et al., 2024; 2023); none of those systems has a link that can vanish mid-transfer.

With the floor at 1 s the routing condition holds for 60-token replies and not for longer ones. 0 of 2,830 replies are routed, carbon falls to 86.5 % of Cloud-Only (11.2 % below the nominal system), mean wait rises from 16.2 to 26.2 ms because of the detour, and the admitted violation count rises from 11 to 13 of 2,738 (0.47 %), still far under the budget. A 6% budget lifts the floor to 8 s, routes every admitted reply, and cuts carbon to 21.5 % of Cloud-Only at 45.8 ms mean wait. Across detours from 10 to 80 ms and energy ratios from 0.1 to 0.5, the cut at 5% stays between 11.1 and 11.3 % while wait moves from 19.4 to 42.5 ms (Figure 9). The dial is the budget, and it is priced in the unit the reader feels.

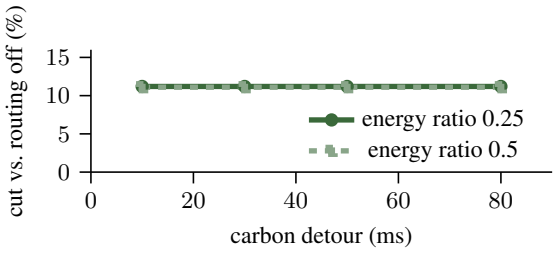


Figure 9: Carbon cut of routing against routing off at the 5% budget, as the detour and the device-to-cloud energy ratio vary; the cut is set by which replies finish inside the floor, not by the detour.

D Calibration Baselines and Additional Sweeps

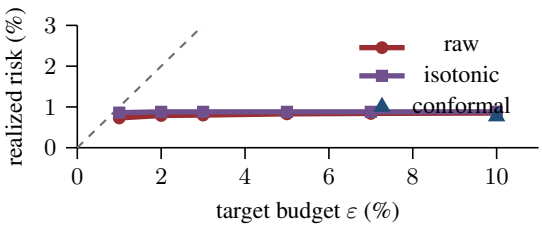


Figure 10: Realized per-second risk of the thresholds of Figure 6 against the diagonal; every method sits far below its budget on this corpus.

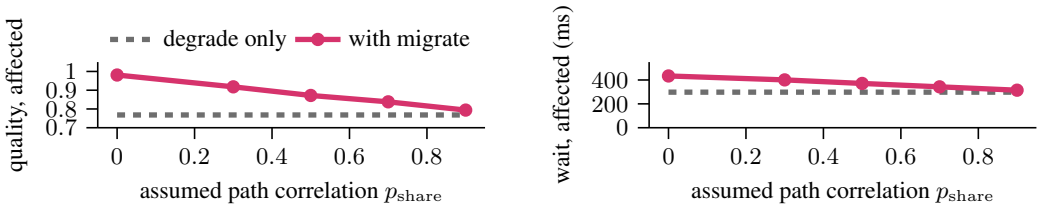


Figure 11: Migration on the 154 replies that needed a hand-off, as the paths are assumed more correlated (one legend for both panels). Left: quality of the subset; migration keeps tokens verified. Right: its mean excess wait; a path switch costs more than a warm drafter's seam. Both decay monotonically.

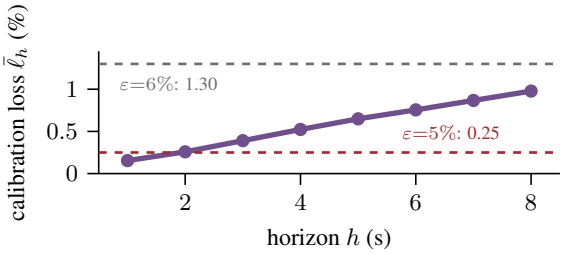


Figure 12: Calibration mean over the 20 traces of the loss of Eq. 4 by horizon, against the corrected bounds $\varepsilon - (1-\varepsilon)/n$ at 5% and 6%; the crossing points are the floors of Figure 2 (bottom left).

Table 4: All five calibrated thresholds and the conformal floor on the nested folds of §5.4 at two budgets: realized per-second risk, fraction of eligible seconds receiving a nonzero floor, and mean floor. For test trace i the ten traces after it in index order train and the next ten calibrate, modulo 21.

	$\varepsilon = 5\%$			$\varepsilon = 10\%$		
Method	Risk (%)	Covered (%)	Floor (s)	Risk (%)	Covered (%)	Floor (s)
raw probability	0.83	99.5	7.59	0.85	99.8	7.87
Platt scaling	0.88	100.0	8.00	0.88	100.0	8.00
temperature	0.85	99.8	7.85	0.87	99.9	7.94
isotonic	0.88	100.0	8.00	0.88	100.0	8.00
histogram binning	0.88	100.0	8.00	0.88	100.0	8.00
conformal	infeasible	0.0	0.00	0.77	99.6	7.08

Table 5: The budget binds. Nested folds as in Table 4, under a 10 kbps interruption definition. "Exceed" counts test traces whose realized risk exceeds the budget among the traces on which the method certified anything. Left block: all 21 traces (10 train, 10 calibrate). Right block: the 12 highest-hazard traces (6 train, 5 calibrate). The conformal floor certifies nothing where the budget is below $1/(n+1)$ (9.1 % and 16.7 %).

		all 21 traces			12 high-hazard traces		
Budget	Method	Risk (%)	Floor (s)	Exceed	Risk (%)	Floor (s)	Exceed
1%	raw	1.10	3.83	9 / 21	1.15	3.13	6 / 12
	temperature	1.58	5.23	14 / 21	1.88	4.77	11 / 12
	isotonic	0.88	2.83	8 / 21	1.00	2.27	6 / 12
	conformal	declines	0		declines	0	
2%	raw	1.08	3.88	3 / 21	1.20	3.24	2 / 12
	Platt	1.89	5.50	10 / 21	2.03	4.54	5 / 12
	isotonic	1.85	5.41	9 / 21	2.05	4.56	6 / 12
	conformal	declines	0		declines	0	
3%	Platt	2.59	7.68	10 / 21	2.76	6.30	5 / 12
	histogram	2.47	7.42	8 / 21	2.79	6.54	5 / 12
	conformal	declines	0		declines	0	
10%	isotonic	2.67	8.00	0 / 21	3.21	8.00	0 / 12
	conformal	0.71	2.00	0 / 21	declines	0	

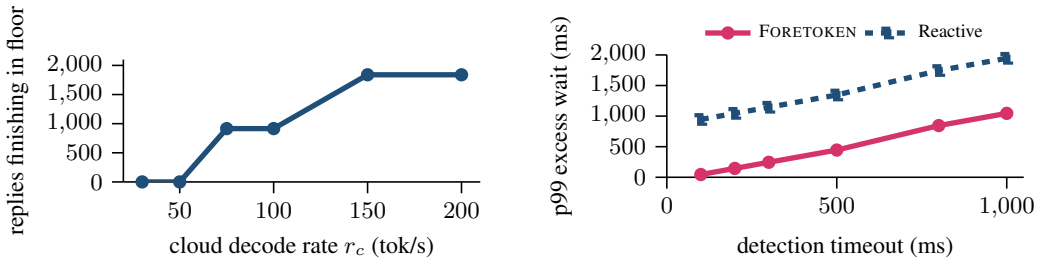


Figure 13: Left: how many of the 2,738 admitted replies satisfy the finish condition inside the 1 s floor as the cloud decode rate varies; only these may take the carbon detour. Right: the p99 tail of FORETOKEN and of Reactive against the detection timeout; both are the timeout plus the drafter's start latency minus t_0.

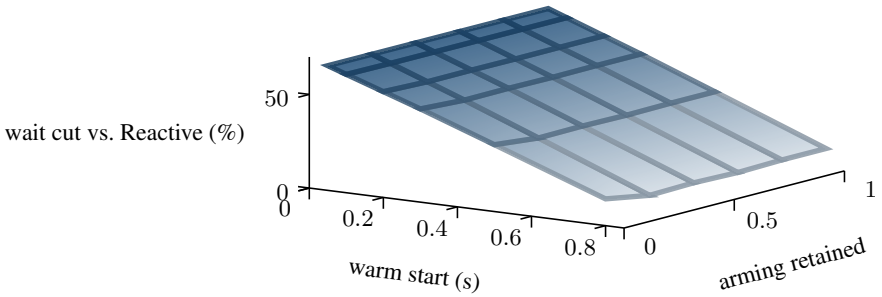


Figure 14: Mean-wait reduction of the pre-emptive variant against Reactive over a 6×6 grid of warm-start cost and the fraction of arming events retained (0 removes the precursor entirely). The sheet is flat along the arming axis and falls along the warm-start axis: the warm drafter, not the precursor, is the lever, the result Table 2 reports in two rows.