Fast or Stealthy, Not Both: The QoS Cost of Evading Residential-Proxy Detection

Shashi Prabh

Ahmedabad University, Ahmedabad

***Abstract*—Residential proxy (RESIP) services resell the IP reputation of real home devices, increasingly sourced from compromised IoT hardware. The BADBOX 2.0 botnet alone spans over ten million devices. Cross-layer round-trip time (RTT) signals collapse under a trivial scheduling attack. Therefore, recent work detects RESIP relays instead by *correlating* the flow between the proxy gateway and the home device against the flow that device relays onward. All of the previous work studies the *detector*. We take the *operator's* view and ask what evasion costs, since a proxy is sold on latency and throughput and every perturbation that hides the relay degrades what the customer paid for. We answer which detector is worth deploying, and how it should be tuned. We model the backconnect relay and its detectors in a packet-level simulator and trace the detectability versus quality-of-service (QoS) Pareto frontier. Defeating the RTT signal is free. Defeating correlation requires smearing traffic across Δ, the time window over which the detector compares the two flows. That costs ≈ 242 ms of added latency to halve recall, and leaves roughly half of all fetches past a 1 s service target. The cost follows a sub-linear power law in the bin width Δ, so the defender's knob has a known price and diminishing returns. A residential proxy can be fast or stealthy, but not both.**

***Index Terms*—Residential proxy, traffic analysis, flow correlation, evasion, IoT botnet, quality of service, network simulation**

I. Introduction

A residential proxy (RESIP) relays a customer's traffic through an IP address that belongs to a consumer ISP allocation, so the destination server sees an ordinary household rather than a datacenter. Because anti-fraud, CAPTCHA, and ticketing systems base reputation on IP address and rarely block residential ranges, a residential exit launders the operator's traffic through the trust of a real home connection. The market is large, and Bright Data alone advertises over 400 million monthly residential IPs across 195 countries [1]. Demand chronically outstrips consent-based supply, which creates economic pressure to source exits by compromise, and cheap, always-on, rarely patched IoT devices are ideal nodes. Supply-chain pre-infection has produced botnets at scale, with the BADBOX 2.0 family alone spanning over 10 million consumer devices [2], [3], and newer families such as Kimwolf continue the growth pattern [4].

Detecting that a residential IP is *relaying*, as opposed to merely classifying it as residential, is an open problem, since IP-intelligence vendors already perform the latter. One natural network signal is a cross-layer RTT discrepancy that exposes the hidden backhaul. Xue et al. [5] point out that this signal is available only to a defender that terminates TLS, since an on-path observer cannot parse the encrypted application exchange. Ali et al. [6] show that the RTT signal is fragile even there, since a trivial scheduling attack drops detector recall from 99% to 8%. The research focus therefore has turned to *flow-correlation* features, which compare the gateway–exit flow against the flow the exit relays onward. The coupling between the two is an architectural invariant and is harder to defeat.

The prior work studies the *detector*, rather than the *operator*. That asymmetry leaves open the question that governs the arms race, namely what evasion costs. The question is answerable because a residential proxy is a product sold on quality of service, and providers compete on advertised sub-second latencies and high success rates. Every perturbation that hides the relay also degrades the bytes the customer paid for. The resulting trade-off has not been measured in prior work.

This paper answers the question, adopting the operator's view as an analytical device. The question it answers is a defender's, namely which detector is worth deploying and how it should be tuned. We build a packet-level simulator of the backconnect relay in which the RTT discrepancy and the gateway–relay correlation *emerge* from an explicit topology and relay logic rather than being assigned by label. We implement five evasion primitives, namely scheduling, inter-packet-delay (IPD) jitter, padding, packet reshaping, and a constant shift, and derive each primitive's effect on the detector and its QoS cost from a single traffic transformation. We trace the detectability–QoS Pareto frontier per detector class.

The paper makes three contributions. The first is, to the best of our knowledge, *the first quantification* of the QoS cost of evading modern flow-correlation RESIP detection, together with an open-source, reusable simulation harness that allows experimentation without any hardware or proprietary traces.[1] The second is *the detectability–QoS Pareto frontier and the detector-upgrade surcharge*, reported at pinned false-positive rates rather than at recall alone. The third is *two bin-width scaling laws* for jitter and constant shift. The cost for both

[1] Available at https://github.com/shashi-prabh/RESIP_WF-IoT_2026.

is a power law in bin-width Δ. Jitter's sub-linearity follows from the correlation samples a bin width affords the detector, making Δ a variance-versus-resolution knob.

II. Related Work

RESIP measurement and characterization. Mi et al. [7] initiated client-side RESIP measurement by renting services and enumerating millions of exit nodes. Subsequent work characterized and classified RESIP traffic, including a recent study of the network traffic of residential proxies [8], but these works map the ecosystem without modeling evasion economics.

RESIP relay detection. The cross-layer RTT discrepancy is the canonical relay signal and underlies both research prototypes and public tooling. *Beyond RTT* [6] is the current state of the art. It points out that RTT detection collapses under a scheduling attack, and instead focuses on gateway–relay *correlation* features that remain robust. We adopt its detector families as our reference and answer the operator's cost question it leaves open.

Traffic-analysis defenses. Flow-correlation attacks and defenses are mature in the anonymity literature, where DeepCorr [9] and DeepCoFFEA [10] correlate flows across a relay. Website-fingerprinting defenses such as WTF-PAD [11] and the constant-rate family analyzed by [12] formalize the *bandwidth–latency cost* of decorrelation. We apply that operating-point methodology to a new adversary model. Here the decorrelating adversary is the proxy operator, and the cost is borne as *degraded product quality*.

IoT-botnet datasets. Public IoT-malware corpora such as IoT-23 [13] capture DDoS, scanning and C2 behavior rather than residential-proxy relaying. Labeled relay traffic is therefore unavailable, and we adopt a simulation methodology grounded in the documented relay architecture.

III. System Model and Problem Formulation

A. Backconnect architecture and vantage points

Consider a RESIP request traversing the four entities of Fig. 1. The *customer* sends it to a provider *gateway*, which tunnels it over a backhaul to a residential *exit*, that is, the compromised IoT device, which finally connects to the destination *server*. While TCP terminates at the exit, TLS is established end-to-end between customer and server, and the exit relays encrypted records without terminating them [6].

We consider two detector vantage points. An *ISP* observes all traffic of the exit device, namely its gateway connections, its relayed connections, and the owner's genuine background traffic, and seeks to label individual connections as relayed or not. A *content provider* observes only the inbound relayed flow at its own server. The adversary is the *operator*, who controls the gateway and the exit, and may perturb the gateway-to-exit and exit-to-server legs, but *cannot* alter the device owner's background traffic and must keep the relayed product performant.

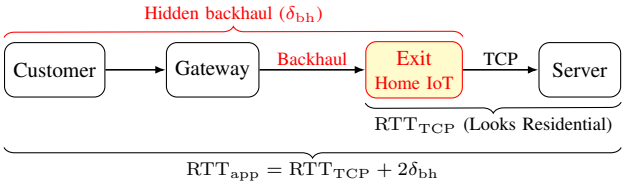


Fig. 1. Backconnect RESIP path. TCP terminates at the exit, so the server-observed $\mathrm{RTT}_{\mathrm{TCP}}$ covers only the short, residential-looking exit to server hop, while the end-to-end TLS $\mathrm{RTT}_{\mathrm{app}}$ also spans the hidden backhaul. Their difference $\approx 2\delta_{\mathrm{bh}}$ is the detection signal.

B. Detection signals

Two metadata signals distinguish a relayed connection. *(i) Cross-layer RTT discrepancy.* With one-way delays δ_{se} (server–exit) and δ_{bh} (backhaul), the server measures $\mathrm{RTT}_{\mathrm{TCP}} \approx 2\delta_{\mathrm{se}}$ but $\mathrm{RTT}_{\mathrm{app}} \approx 2(\delta_{\mathrm{se}} + \delta_{\mathrm{bh}})$, so the discrepancy $\delta_{\mathrm{RTT}} \approx 2\delta_{\mathrm{bh}}$ exposes the backhaul. A direct connection has $\delta_{\mathrm{RTT}} \approx 0$. *(ii) Flow correlation.* Because the gateway forwards exactly the relayed bytes, the gateway flow and the relayed flow are correlated. The detector bins both flows at width Δ (e.g., $\Delta = 0.1$ s), z-scores each series, and summarizes their element-wise product. Background traffic, not carried by the gateway, is uncorrelated.

C. Evasion primitives

The operator perturbs the relayed flow with five primitives, each a single transformation with a tunable intensity. *Scheduling* re-times one handshake packet to mask δ_{RTT}. *IPD jitter* delays each packet by a half-normal draw of scale σ and thereby smears the volume series across bins, the delay being one-sided since a packet cannot be sent early. *Constant shift* delays the whole relayed series by a fixed τ, which translates it rather than smearing it, and therefore misaligns it against a correlator that compares the two flows. *Padding* injects chaff bytes at independent times, in a volume p times the payload. *Reshaping* splits packets into m fragments. The correlation feature z-scores each binned series, so *uniform* size inflation is normalized away, and decorrelation therefore requires perturbing packet *timing* through jitter or adding bytes at *independent times* as chaff. Enlarging packets does not suffice.

D. QoS cost model

We model *per-fetch* latency, which is what the customer experiences and the product is sold on. A response of S bytes (N packets) is relayed from exit to gateway over the slow, asymmetric residential *uplink*, which is the relay bottleneck. Let μ be the uplink speed. Let $r = 2(\delta_{\mathrm{se}}+\delta_{\mathrm{bh}})$ be the end-to-end RTT and $T_{\mathrm{setup}} = 2r$ the handshake cost, one round trip for TCP and one for TLS. We assume k TCP slow-start rounds, with k fixed at $k=3$. The baseline per-fetch latency is

$$T_0 = T_{\mathrm{setup}} + \max\big(S/\mu,\ k\,r\big). \tag{1}$$

Each primitive maps to a cost described as follows.

Scheduling re-times a single handshake packet, which leaves the byte series untouched and adds no delay to the transfer, so we charge it nothing. Its effect is confined to the RTT features, where it pushes $\delta_{\mathrm{RTT}} \to 0$, so a detector resting on that discrepancy is evaded at no cost to the product.

IPD jitter delays packet i by $d_i = \sigma|Z_i|$, where $Z_1, \ldots, Z_N$ are independent standard normal variates, so each delay is half-normal with scale σ. We charge the completion time the expected maximum of those N draws,

$$\ell(\sigma) = \sigma\, c_N, \tag{2}$$

$$c_N \;=\; \mathbb{E}\Big[\max_{i\le N} |Z_i|\Big] \;=\; \int_0^\infty \big(1 - \mathrm{erf}(x/\sqrt{2})^N\big)\, dx, \tag{3}$$

a pure *latency* cost. This is an upper bound rather than an identity, since the last byte arrives at $\max_i(t_i + d_i)$, where t_i is packet i's unperturbed send time, rather than at $t_N + \max_i d_i$. We evaluate c_N exactly rather than using the familiar asymptote $\sqrt{2\ln N}$, which is the *normal* extreme-value limit and overstates c_N.

Padding inflates bytes to $B' = (1+p)S$, a *bandwidth* cost of p. It becomes a latency cost only when serialization overtakes the round-trip-bound term, that is, when B'/μ exceeds $k\,r$.

Constant shift delays every packet equally, so the completion time moves by exactly τ. It adds no bytes and no spread, which makes it the one primitive whose cost is the displacement itself rather than a multiple of it.

Reshaping splits packets into m fragments. It adds $(m-1)Nh$ bytes for per-packet overhead, $h = 70$ B. Goodput loss is $1 - T_0/T_{\mathrm{ev}}$, where T_{ev} denotes the evaded completion time.

E. The operator's problem and the shape of the frontier

Let D be a deployed detector and θ the evasion configuration. The operator solves

$$\min_{\theta}\; R(\theta; D) \quad \text{s.t.} \quad C(\theta) \le \beta, \tag{4}$$

where R is detector recall and C the QoS cost under budget β. The achievable set $\{(C(\theta), R(\theta; D))\}$ defines a Pareto frontier $\mathcal{P}(D)$. To decorrelate a Δ-binned correlation, jitter must spread a packet's mass over a comparable timescale, so that $\sigma \sim \Delta$. Combining with (3), the evasive latency is

$$\ell \;=\; \sigma^*(\Delta)\, c_N, \qquad \sigma^*(\Delta) = \Theta(\Delta), \tag{5}$$

where $\sigma^*(\Delta)$ is the minimum standard deviation required to decorrelate the Δ-binned correlation. Hence the cost grows with the detector's bin width.

Equation (5) is an *achievable* bound within the half-normal jitter family, not a fundamental lower bound over all evasions. The cost is the *maximum* of the delay draws whereas decorrelation depends on their *spread*, so a delay distribution optimized to minimize $\mathbb{E}[\max_{i\le N}|Z_i|]$ subject to a per-bin mass-spreading constraint would decorrelate more cheaply. We also do not model the reordering that independent per-packet delays induce, which would raise the operator's cost further through duplicate acknowledgements and retransmission.

IV. Simulation Pipeline

Simulator. We implement a packet-level simulator of the four-entity path (Fig. 1) in which the two detection signals *emerge* from the topology. The RTT discrepancy follows from $\delta_{\mathrm{bh}} > 0$ on relayed connections, and the correlation from constructing each window's gateway flow as the sub-bin-lagged sum of its relayed byte series. Traffic is generated as a set of independent device *windows* at realistic low concurrency. *Beyond RTT* reports ≈ 110 relayed connections per two-hour capture with a *mean* duration of 28.75 s, so fewer than one is active at a time on average and each relayed connection dominates its gateway flow. Relayed-connection size and duration follow the reported distributions, while gateway flows carry 60 s heartbeats and cross-traffic noise. Relayed and background connections share identical size, duration and packet marginals, so a detector's discriminative power is attributable only to the two signals under study.

Detectors. We reconstruct three feature families from [6]. These are `RTT`, which uses cross-layer timing only, Traffic Analysis features (`TA`), which adds volume and structural summaries over the first packets, and Flow Correlation features (`Corr`), the binned and z-scored gateway–relay correlation. Classifiers are gradient-boosted trees. Scheduling is essentially free, so we assume it is always deployed and *harden* the detector against it by training on a mixture of clean and scheduled traffic, so it cannot over-rely on the RTT feature and must learn the correlation signal.

Operating points. Recall alone does not define an operating point, so we pin each detector's decision threshold at a target false-positive rate on *clean* traffic and hold it fixed while the operator evades. The headline surcharge is read at each detector's own decision threshold, whose calibrated rate on the same clean draw is 3.2×10^{-3} for TA+Corr (5.1×10^{-3} for Corr, 9.5×10^{-2} for RTT); the two pinned rates in §V-B bracket it. The operator cannot touch the device owner's background traffic, so the negative score distribution is identical across evasion configurations and the FPR is unchanged by construction. That is what makes the *cost to halve recall at* $\mathrm{FPR} = 10^{-3}$ well defined. Thresholds are calibrated on an independent clean draw of 14,285 background connections, redrawn per seed, so the reported bands propagate threshold-estimation error. The 10^{-3} threshold rests on only the 15th highest negative score, against the 143rd at 10^{-2}, and hence it should be read as the coarser of the two.

Reading the cost. The main cost we consider is the added latency at which recall crosses 50%, linearly interpolated between the two bracketing points of a Δ-relative intensity grid whose spacing holds the bracket within 4–17% of the cost across the swept range. This is therefore an interpolated quantity rather than a measured grid value. Table I lists the simulator settings.

Link and workload parameters. We calibrate four access-network regimes from public broadband measurement data [14], [15]. The four bandwidth (down/up) and delay settings, each with a geographically distributed backhaul, are:

- Fiber: 300/300 Mbps, idle RTT 7–14 ms
- Cable: 200/20 Mbps, 12–24 ms
- DSL: 15/1.5 Mbps, 23–34 ms
- Fixed-wireless: 50/10 Mbps, 20–40 ms

The resulting per-fetch baselines of 733–949 ms (Fig. 5) are consistent with the expected sub-second per-request latencies. We do not treat this as a validation of the QoS model, since the advertised envelope spans roughly an order of magnitude and clearing it is a weak test that most plausible parameterizations would pass.

Validation. The pipeline reproduces the qualitative result of [6]: the free scheduling attack drops `RTT` recall from 100.0% to 9.9% while `Corr` is untouched at 97.8%, since retiming one handshake packet never enters the volume binning, and `TA+Corr` survives at 99.0%.

Reference-detector fidelity. Our `Corr` features reproduce the released [6] extractor's core computation, namely 0.1 s volume binning, per-series z-scoring, and the element-wise product. They do not reproduce its windowing, since the released code correlates over the *first* 20 packets whereas we correlate over the whole connection. Under a parity check at the released windowing and its full nine-statistic summary the qualitative results are unchanged. The narrower window sees fewer bins, however, and paired on the six seeds the cost to halve `TA+Corr` recall falls by $34 \pm 5\%$. The surcharge is thus sensitive to the detector's windowing, which is a further reason to read the scaling law of Sec. V as the transferable result rather than any single millisecond figure.

TABLE I
SIMULATOR SETTINGS. WORKLOAD MARGINALS FOLLOW [6].

Parameter	Value
Observation windows / window length	400 / 120 s
Relayed, background conns. per window	Poisson 2.0, 6.0
Conn. size drawn (lognormal median)	63 KB
Conn. duration (lognormal median)	9 s
Packets per fetch N (median / mean)	54 / 74
Packet size, down / up	1200 / $\mathcal{U}(80, 300)$ B
Per-packet header h (reshaping)	70 B
TCP slow-start rounds k	3
Gateway heartbeat / noise	60 s, 150 B / 15%
One-way last hop δ_{se}	$\mathcal{U}(5, 60)$ ms
One-way backhaul δ_{bh}	$\mathcal{U}(15, 120)$ ms, mean 67.5
(swept over its mean)	12.5–130 ms
Detector correlation bin Δ	100 ms (swept 25–800)
Jitter grid σ	12.5–600 ms, 12 points
(Δ-relative)	0.2–2.0 Δ, 16 points
Default uplink μ	12 Mbps
Train/test split, seeds	70/30, 6 seeds

V. RESULTS

A. Two ways to break the correlation

With scheduling always on, RTT detection is already defeated, so the operator must defeat correlation. Two primitives do it, and Fig. 2 plots both against what the operator pays rather than against their own intensity, which is the only axis on which they are comparable. As IPD jitter σ increases, recall stays near 98% until σ approaches the detector's 0.1 s bin width, then collapses through 50% at $\sigma \approx 0.10$ s. Equation (5) fixes only the *order* of that threshold. At $\Delta = 100$ ms, the ratio $\sigma^*/\Delta = 0.96$. Latency grows in proportion to σ by Eq. (3), so halving recall costs ≈ 242 ms and pushing recall to $\approx 10\%$ costs over 500 ms.

A constant shift reaches the same 50% line for ≈ 62 ms, a quarter of the price. It is cheaper for a structural reason rather than a subtle one. The detector compares the two flows at zero lag, so a translation of the relayed series misaligns it outright, and the operator pays the displacement once instead of paying the expected *maximum* of N per-packet draws.

The shift arm of Fig. 2 stops at one bin width, and the truncation is deliberate. Our packet times are drawn independently within a connection (Sec. III), so the binned volume series is close to white. Shifting by a full bin therefore lands on an uncorrelated stretch and collapses the correlation feature to its floor. Below one bin the decay is the transfer of byte mass between adjacent bins, which does not depend on that assumption. Beyond it, the curve would be reporting our arrival process rather than the primitive. Real traffic is bursty, retains correlation across several bins, and would leave a higher residual. Therefore, τ^* measured here is a *lower* bound and the four-fold advantage over jitter is an upper bound on the advantage. We therefore claim the shift only as the cheaper primitive against a bin-based zero-lag correlator, not as a transferable constant. We discuss this further in Sec. VI.

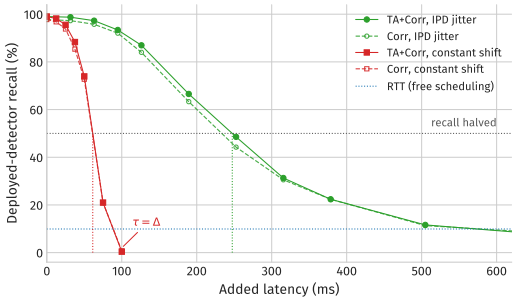


Fig. 2. Primitives against the QoS they cost. A constant shift halves `TA+Corr` recall for a quarter of what jitter costs. Its arm stops at $\tau = \Delta$, past which our independent arrivals leave nothing to correlate and the curve would report the packet model rather than the primitive.

B. The detectability–QoS frontier

Fig. 3 plots recall against both cost axes for all evasion configurations. RTT detection is either perfect or, under scheduling, free to defeat, whereas correlation degrades only after hundreds of milliseconds. Padding alone does not suffice, since even 200% byte overhead leaves correlation recall at $84 \pm 9\%$. Here z-scoring normalizes uniform inflation, chaff decorrelates only weakly, and reshaping alone is weaker still.

The contrast defines a *detector-upgrade surcharge*. The cost to halve recall is ≈ 0 ms for the RTT detector under scheduling, against ≈ 242 ms for the correlation detector. A defender who upgrades from timing to correlation therefore forces the operator off the free corner of the frontier onto an

expensive one. Against the calibrated per-fetch baselines of ≈ 0.73–0.95 s (Fig. 5), that surcharge is a 26–33% latency penalty merely to *halve* detection, and proportionally more to suppress it further.

Operating points. Recall is also read at thresholds pinned to a fixed false-positive rate rather than reported alone. The surcharge is 314 ± 30 ms at $\mathrm{FPR}=10^{-2}$ and 198 ± 19 ms at 10^{-3}, the stricter threshold being cheaper to evade because it starts closer to the 50% line. The 242 ± 27 ms for 3.2×10^{-3} operating point sits between the two. Hundreds of milliseconds on the latency axis is therefore the picture across the operating points a deployment would plausibly choose.

Mixed and variable primitives. The harness composes primitives, and we sweep the combinations of [6] in Table II over the same six seeds, reporting each as a paired difference against the pure-jitter arm of the same seed. That arm costs 249 ± 29 ms, matching the headline. Two results qualify the single-primitive picture. First, bandwidth is a *partial substitute* for latency, since adding 100% chaff on top of jitter cuts the cost to halve recall by 101 ± 9 ms, with the same sign on every seed we tested. Padding is neutralized by z-scoring only when it is the *sole* perturbation. The substitution stays partial, because no bandwidth-only configuration we tested reaches 50% recall at any overhead. Second, reshaping helps the detector rather than the evader, since splitting packets populates more bins and thereby makes the series easier to track. Randomizing σ per connection costs 24 ± 13 ms more than a matched mean σ, since connections drawn near $\sigma=0$ stay fully detectable.

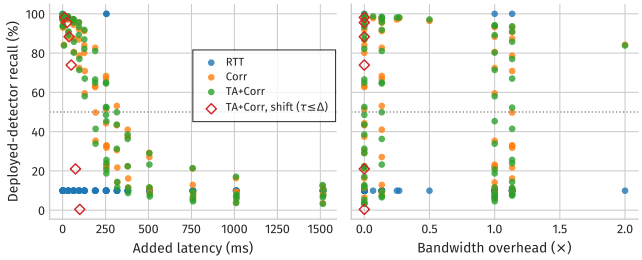


Fig. 3. Detectability–QoS frontier, in which lower-left is the evader's goal. Left: added latency. Right: bandwidth overhead. Open diamonds are the constant-shift family, which reaches the lower-left corner well before any other primitive.

TABLE II
COMBINED ATTACKS (SC SCHEDULING, TP PADDING, PR RESHAPING, IPD JITTER). MEAN OVER SIX SEEDS, RECALL S.D. 3–12 POINTS. BW IS ADDED BYTES AS A PERCENTAGE OF THE UNEVADED PAYLOAD, AT THE TABLE I LINK.

Setting	Lat. (ms)	BW (%)	Corr (%)	TA+Corr (%)
SC+TP	2	100	93.7	94.2
SC+TP+PR	3	113	93.0	93.6
SC+IPD	252	0	44.2	48.6
SC+PR+IPD	253	13	68.4	65.0
SC+TP+IPD	255	100	21.9	20.7
SC+TP+PR+IPD	255	113	33.0	25.6

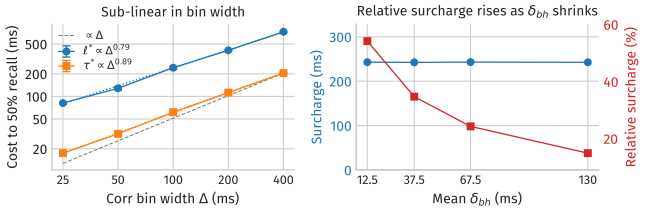


Fig. 4. Sensitivity of the surcharge. Left: on log–log axes both primitives are sub-linear power laws in the bin width (grey guide: slope 1); points are the six-seed means of Table III with standard error, dotted lines the weighted fits. Right: the surcharge is fixed by Δ (blue), so a shorter backhaul shrinks only the baseline it is measured against and the relative surcharge rises (red).

C. The bin-width scaling law

Both primitives cost *sub-linearly* in the detector's correlation bin width Δ (Table III). Weighted log–log fits over $\Delta\in[25,400]$ ms, six seeds per point, give

$$\ell^* \propto \Delta^{0.791\pm0.012}, \qquad \tau^* \propto \Delta^{0.891\pm0.005}, \tag{6}$$

at χ^2/dof of 0.95 and 0.99. Coarsening bins therefore raises the operator's price with *diminishing returns*. For jitter the mechanism is sample count, since a 9 s connection yields 360 correlation samples at $\Delta=25$ ms against 22 at 400 ms, so finer bins give a lower-variance estimate that takes relatively more perturbation to break. The shift exponent is weaker evidence. Under independent arrivals τ^* reports the transfer of byte mass between adjacent bins as much as it reports the primitive. The ratio τ^*/Δ implied by Table III falls from 0.72 at $\Delta=25$ ms to 0.51 at 400 ms rather than holding fixed, and bursty arrivals would raise it further.

The other sweeps move the cost little. Invariance to uplink and backhaul is an algebraic identity rather than a measurement, since jitter adds no bytes and hence $T_{\mathrm{ev}}-T_0=\sigma c_N$ holds exactly. What the backhaul changes is the *denominator*, so across a $10\times$ range of means (Table I) the same surcharge grows from 15% to 54% of what the customer was buying (Fig. 4), and the operator's two levers for a better product, fast access and short backhaul, both make it hurt more in relative terms. Concurrency moves the cost modestly and in the direction opposite to dilution, and the ≈ 1.8 relayed connections per 120 s window of [6] sit inside the swept range.

TABLE III
BIN-WIDTH SCALING: COST TO HALVE TA+CORR RECALL, MEAN OVER SIX SEEDS WITH STANDARD ERROR OF MEAN.

Bin width Δ (ms)	25	50	100	200	400
IPD jitter ℓ^* (ms)	81 ± 2	128 ± 7	242 ± 11	413 ± 7	729 ± 19
Shift τ^* (ms)	18 ± 0.2	32 ± 0.9	62 ± 1.6	113 ± 1.1	205 ± 2.9

D. Calibrated access regimes

We re-run the frontier under the four calibrated regimes (Fig. 5). Three main results are: (i) the decorrelation surcharge is access-technology invariant, at 242–246 ms across all four, a spread of 4 ms against a ±27 ms seed band, since the surcharge is set by the bin width rather than by the link (Eq. (5)).

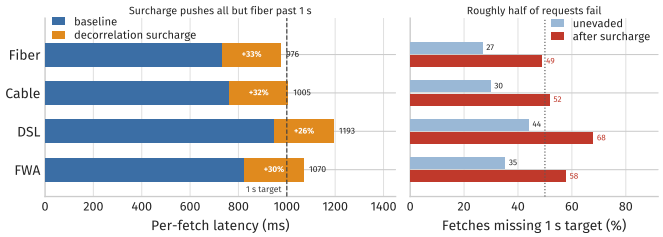


Fig. 5. Calibrated regimes. Left: per-fetch baseline (blue) plus the near-constant surcharge (orange), which carries every regime except fiber past the 1 s target (dashed), and fiber to within 25 ms of it. Right: > 1 s share of requests before and after surcharge.

(ii) The performance cost of padding is *not* invariant. On the slow DSL uplink that bandwidth turns into latency, adding 312 ms and 19% goodput loss at 100% padding, whereas on fiber it is free. The relative surcharge is therefore highest on the fiber and cable exits that operators prefer, so the regime where bandwidth is cheapest to spend is the regime where it buys the least relief. (iii) Finally, milliseconds understate the harm. The baseline already sits near the 1 s mark that commercial services advertise against. So the surcharge raises the share of fetches missing that target from 27–44% to 49–68%. An operator evading a correlation detector therefore fails its advertised service level on roughly half of all requests.

VI. Discussion and Limitations

Implications. Defenders should deploy correlation-based detection in preference to timing-based detection, since the latter is evaded for free, and treat the correlation bin width as a tunable security parameter. Coarsening bins raises the operator's price sub-linearly, so there is a point past which the resolution given up is not repaid. The costs are upper bounds, since the surcharge is what the half-normal jitter family charges and an optimized delay distribution would decorrelate more cheaply. The paradox that arises due to correlation needing both flows at once is that the ISP that can turn the knob carries little of the loss while the platforms that want RESIP detection cannot correlate at all.

Limitations. Our results come from a simulator that is not yet validated against real relay captures, and we model only *fixed-line* exits, whereas providers increasingly recruit exits via mobile-app SDKs [6]. Our arrivals are independent within a connection, so the binned series holds almost no autocorrelation and a one-bin shift decorrelates it completely. Bursty arrivals would keep the two series partly aligned past a bin and raise τ^* by the burst correlation length, and if that residual exceeds the operating point a shift would not halve recall at any price, so the shift is established as the cheaper primitive only against near-white input.

Ethics. This work is defensive, and it uses no human-subject data and no live botnet.

VII. Conclusion and Future Work

This work took the operator's view of residential-proxy detection and priced evasion. Defeating the RTT signal is free. Defeating correlation is not, and requires smearing traffic across the detector's binning interval, at ≈ 242 ms of added latency to halve recall and roughly half of all fetches past a 1 s target. The cost is a power law in the bin width, as $\Delta^{0.79}$, which is a tuning knob for the defender with a known price and diminishing returns. Interesting future work includes validating the model against `netem`-emulated traffic, bursty arrivals and solving for the delay distribution that minimizes the expected largest per-packet delay, at fixed decorrelation.

Acknowledgment

The comments of anonymous reviewers helped in improving the quality of the paper. The author is grateful for their time. This work was supported by Ahmedabad University grant number URBSEASE20A6/SUG/20-21/03-SP-08.23.

References

[1] "Bright data residential proxies," 2026, https://brightdata.com/proxy-types/residential-proxies.
[2] HUMAN Security Satori Threat Intelligence, "BADBOX 2.0 targets consumer devices with multiple fraud schemes," 2025, https://www.humansecurity.com/learn/blog/satori-threat-intelligence-disruption-badbox-2-0/.
[3] Federal Bureau of Investigation, "BADBOX 2.0 botnet infects internet-connected devices," 2025, public Service Announcement, IC3 Alert I-060525-PSA.
[4] QiAnXin XLab, "Kimwolf: A residential-proxy botnet exploiting exposed ADB interfaces," 2025, threat disclosure.
[5] D. Xue, R. Stanley, P. Kumar, and R. Ensafi, "The discriminative power of cross-layer RTTs in fingerprinting proxy traffic," *Proceedings 2025 Network and Distributed System Security Symposium*, 2025.
[6] T. Ali, S. Yoosuf, M. Rabhi, M. Al-Sabah, and H. Yun, "Beyond RTT: An adversarially robust two-tiered approach for residential proxy detection," in *Network and Distributed System Security Symposium (NDSS)*, 2026.
[7] X. Mi, X. Feng, X. Liao, B. Liu, X. Wang, F. Qian, Z. Li, S. Alrwais, L. Sun, and Y. Liu, "Resident evil: Understanding residential IP Proxy as a Dark Service," in *2019 IEEE Symposium on Security and Privacy (S&P)*, 2019, pp. 1185–1201.
[8] R. Huang, D. Zhao, X. Mi, and X. Wang, "Shining light into the tunnel: Understanding and classifying network traffic of residential proxies," *arXiv preprint arXiv:2404.10610*, 2024.
[9] M. Nasr, A. Bahramali, and A. Houmansadr, "DeepCorr: Strong flow correlation attacks on Tor using deep learning," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018, pp. 1962–1976.
[10] S. E. Oh, N. Mathews, M. S. Rahman, M. Wright, and N. Hopper, "DeepCoFFEA: Improved flow correlation attacks on Tor via metric learning and amplification," in *2022 IEEE Symposium on Security and Privacy (S&P)*, 2022, pp. 1915–1932.
[11] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," in *European Symposium on Research in Computer Security (ESORICS)*, 2016, pp. 27–46.
[12] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg, "A systematic approach to developing and evaluating website fingerprinting defenses," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014, pp. 227–238.
[13] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IoT-23: A labeled dataset with malicious and benign IoT network traffic," in *Zenodo*, 2020, https://www.stratosphereips.org/datasets-iot23.
[14] Federal Communications Commission, "Measuring broadband america: Fixed broadband report," 2021, https://www.fcc.gov/reports-research/reports/measuring-broadband-america.
[15] Measurement Lab, "The M-Lab NDT data set," 2026, https://www.measurementlab.net.