Outages Are Inevitable – Meltdowns Are Optional

By **Erik Rudin**, VP of Marketing and Corporate Strategy

When AWS stumbles, the Internet trips. Yesterday's multi-hour disruption rippled through banks, retailers, airlines, gaming platforms, and popular apps many of us rely on every day. Beyond consumer inconvenience, the business impacts are huge. As Reuters reported, "For major businesses, hours of cloud downtime translate to millions in lost productivity and revenue."

This wasn't a cyberattack. It was an operational failure in foundational plumbing. AWS's Health Dashboard and subsequent reporting pointed to DNS resolution trouble in the US-EAST-1 region (affecting the DynamoDB API), alongside a malfunction inside the subsystem that monitors the health of network load balancers (LB) – a Layer-4 building block many services depend on. That combination created a cascading effect across services and geographies.

When the health subsystem falters, load balancers can enter "fail-open" behavior and route traffic to unhealthy targets. Mix that with flaky DNS to a core dependency (DynamoDB) and you get retry storms, timeouts, and a blast radius that crosses teams and toolchains.

Building Resilience: Strategies to Minimize Outage Impact

We're respectfully cynical about "it can't happen here." It can. And when it does, the top priorities are rapid detection, tight blast-radius control, and clear proof that help teams act with confidence. That starts with trustworthy data.

Packets are the ground truth about what's happening on your network. If you can see every conversation – client to DNS, app to database, hop-by-hop through routers, firewalls, and load balancers – you can quickly separate symptoms from the cause. That's the backbone of cPacket's unified network observability approach: full-fidelity packet visibility, distilled into verifiable insights that explain what happened, where, and when. We surface explainable, AI-assisted insights so your engineers can decide why and what to do next.

A practical playbook you can run now

Here's hype-free checklist we recommend (and implement with customers) to turn multi-hour incidents to minutes-long events:

- **Instrument the dependency chain from end-to-end.** Tap both sides of critical choke points – DNS resolvers/forwarders, load balancers (ingress and egress), gateways, and the services they front. That lets you confirm if a failure is upstream (DNS), mid-path (LB health), or downstream (targets) in seconds, not hours.
- **Baseline the boring.** Continuously learn "normal" DNS response codes/latency and LB health-check patterns per service and per site. When entropy spikes or checks go inconsistent, raise a high-fidelity alert that points to the exact hop, not just "the app is slow."
- **Correlate symptoms to a single root cause.** Tie client errors (timeouts, resets) to upstream DNS anomalies and LB mode changes. If your LB flips into fail-open while DNS is flapping, your platform should let you know – plainly avoiding finger-pointing across teams.
- **Prove it with packets.** When leadership asks, "are we safe to fail back?" you'll have packet-level evidence – PCAPs, timelines, and per-hop metrics – to validate recovery and close the incident with confidence.
- **Practice blast-radius drills.** Regularly rehearse DNS/LB impairment scenarios. Degrade a resolver, flip health-check behavior in staging, confirm your detections fire within seconds and route engineers to the right hop immediately.
- **Design for graceful degradation.** Where possible, avoid single-region and single-resolver dependencies for critical paths. If US-EAST-1 stutters, your control plane shouldn't take the customer experience with it.

Closing thought (and a nudge)

Yesterday reminded everyone that cloud concentration risk is real – but so is the payoff when you invest in continuous, packet-based observability. If you want to pressure-test your DNS/LB monitoring posture – or see how explainable AI insights can shorten your MTTD/MTTR on days like this – let's run a targeted review on your most critical paths.

Resources:

AWS Service Health status & history: https://health.aws.amazon.com/health/status

Reuters: https://www.reuters.com/business/retail-consumer/amazons-cloud-unit-reports-outage-several-websites-down-2025-10-20/