RAPAX

TECHNICAL WHITE PAPER

The Last Two Percent

Hunting the Network Gremlins That Break Your SLA

The costliest outages **fix themselves before anyone arrives**, then come back. Rapax 2.0 finds them in the alert history you already have and tells you when the next one is due.



Shawn Ennis
Founder & CEO, Rapax
rapax.app/evaluate · cal.com/shawn-ennis

Chronic alert detection in Rapax 2.0
Written for: CTO, VP Engineering, VP Operations
Telecom operators and enterprise / MSP NOCs
Version 1.0 · October 2026

2.5%
of links caused more than half of all failures in one Tier 1 backbone [B]

25.9 s
a month: the whole downtime budget of a five-nines service

17–20%
of telecom dispatches are estimated to find no fault [C]

WHAT THIS PAPER CLAIMS, AND WHAT IT DOES NOT

This paper describes chronic alert detection as it ships in Rapax 2.0. What 2.0 does not do is listed as a limit, not promised as a feature. Screens and lab figures come from a Rapax lab system.

Every figure is graded. Industry figures are attributed in the Sources section; estimates are labelled and show their assumptions. Where a widely repeated number could not be traced to a primary source, we say so. This edition contains no Rapax production figures for chronic detection.

© 2023–2026 Rapax, a Citus Technologies, LLC company.

Executive summary

Most outages are straightforward. A fiber gets cut or a power supply fails. An alarm fires, a ticket opens, and someone fixes it. Your tools, runbooks and people are built for those outages, and they handle them well.

The expensive outages are the ones that come and go. A router drops for half an hour most Friday afternoons. A link flaps a few times a day and is clean every time someone looks. By the time a technician arrives, the service is back up and nothing looks wrong. The modem gets replaced on a hunch, the ticket closes, and next week it happens again. These are network gremlins, and they are where five-nines commitments quietly fail.

Three things make gremlins costly:

- **They are concentrated.** Published studies of production networks find a small minority of links generating most failure events. In one Tier 1 backbone, 2.5% of links accounted for more than half of all failures.
- **They cost far more than their size suggests.** They bring repeat truck rolls, parts swapped on guesses, engineering hours and SLA credits. In contracts with chronic-outage clauses, they also give the customer the right to walk away.
- **Noise reduction hides them.** Modern tools are tuned to make self-clearing alerts disappear. A self-clearing alert that keeps coming back is exactly what a gremlin looks like.

Calling chronic outages “noise” is a recipe for SLA violations, a low-quality network and upset customers.

Rapax 2.0 treats them as signal. Chronic detection runs every five minutes over eight weeks of alert history. It applies three detectors (flapping, recurring and periodic) and raises a ChronicAlert that groups every occurrence under one parent.

When the pattern follows a fixed weekly or daily schedule, Rapax states when the next window is expected and warns you before it opens. A technician can then be standing in front of the equipment when the gremlin appears. When the pattern doesn’t follow a schedule, Rapax tells you where to look and gives the technician the complete history in one place.

You don’t have to wait eight weeks to find out what is hiding in your network. Alert history from your current tools can be loaded through a custom import, set up for your source during onboarding. Send us eight weeks of alert history, and the free evaluation will show you your gremlins.

1. The cleaner who broke the SLA

In 2014 I worked a problem at a Tier 1 U.S. carrier that every NOC veteran will recognize. A remote router at a business customer's site went down for about thirty minutes on most Friday afternoons, sometime between 2 and 5 p.m. local time. It didn't happen every Friday: often enough to breach the SLA, rarely enough to look random.

The response followed the standard pattern. The alarm fired, a ticket opened and a technician was dispatched. By the time the truck arrived, the router was back up. The interfaces were clean, and the logs showed a restart and nothing else. We replaced the modem, and it happened again. We escalated to the vendor, who asked us to reproduce it, which we couldn't do on demand. The customer, reasonably, wanted credits.

This went on for months, if not years. In the end we found the cause in the most expensive way there is. We put a technician in the room every Friday afternoon, for weeks, watching the router. The answer was the cleaning crew, who unplugged the router to vacuum.

Everything makes sense after the fact. The fix was a sign over the outlet, and we billed the customer back for the SLA credits. We still came out behind, because the real cost was never the credits. It was the dispatches, the hardware swapped on guesses and the engineering time. It was also months of customer confidence spent while nobody knew why.

Two things about that story matter for the rest of this paper. First, the clue was in the data the whole time: a Friday-afternoon outage that repeated week after week. Second, finding it was almost all the work. Once someone was in the room at the right time, the fix took minutes.

~30 min
down, on most Friday afternoons between 2 and 5 p.m.

Months
of dispatches, swapped hardware and vendor escalations

1 sign
over the outlet. The fix, once someone saw the cleaning crew

2. A few links, most of the pain

Ask an operations leader where their outage minutes come from, and you will usually hear about the big events: the fiber cut, the power failure, the bad software push. Those dominate the incident reviews, but they aren't where most failure *events* come from.

Published research on production networks is consistent on this point:

- In Sprint's IP backbone, 2.5% of links accounted for more than half of all failures. This figure comes from Markopoulou et al., as cited by UC San Diego researchers. [B]

- UCSD's study of the CENIC research and education network covered five years and more than 200 routers. One flapping link generated more than 33,000 short failures in six months, 93% of all failures on its sub-network. [B] The cause turned out to be an optical device damaged during an earlier fiber repair, and it was hard to diagnose.
- Across CENIC customer links, the average link failed 302 times a year while the median link failed 20 times. That gap means a few chronic links dominate the count. [B]
- The same researchers describe another link that failed only a few times a day. At least one diagnosis attempt failed before faulty hardware was identified. [B]

From the field, my estimate is that roughly 2% of a network's faults are chronic gremlins of this kind, and that they consume most of the time spent finding problems. That is an operator's estimate, not a measured statistic. No published study we found measures intermittent faults' share of time to resolution. [C]

Every concentration study we reviewed supports that direction, and so does regulatory data on repeat troubles. The FCC defines a repeat trouble as one that recurs or stays unresolved within 30 days, and uses the repeat rate to judge how well a carrier fixes problems the first time. Texas regulators expect a carrier to take corrective action when more than 22% of trouble reports are repeats. One small Texas carrier reported repeat rates of 28–31% in a single quarter. [B]

A small share matters this much because of the arithmetic of availability.

Table 1. Allowed downtime by availability target

Availability	Per year	Per 30-day month
99%	87.6 hours	7.2 hours
99.9%	8.76 hours	43.2 minutes
99.99%	52.6 minutes	4.3 minutes
99.999%	5.26 minutes	25.9 seconds

At 99%, a gremlin is an irritation. At 99.999%, the monthly budget is shorter than a single router reboot. A device that flaps for three seconds a hundred times a year uses up 95% of a five-nines service's annual budget on its own.

25.9 s

a five-nines service's whole downtime budget for a 30-day month

95%

of a five-nines year used by 100 three-second flaps

302 vs 20

average vs median failures per CENIC link per year [B]

There is an honest tension here. Long outages still dominate total downtime minutes, and the median repair time on CENIC links was measured in seconds. Gremlins dominate the *count* of disruptions. The count is what customers remember, what repeat-trouble metrics track, and what many contracts measure.

3. Noise or signal?

Every modern NOC fights alert fatigue, and the industry's answer is noise reduction: deduplicate, correlate, suppress. Much of that is good engineering. But these techniques share an assumption that should worry anyone who owns an SLA: an alert that clears itself is an alert nobody needs to see.

The assumption is explicit. One widely used incident-response platform markets machine learning that detects and pauses transient alerts that have historically resolved themselves. It advertises noise reduction of up to 98%. [C] vendor claim

Operators do the same by hand. A Microsoft Research study of data-center network failures found that link flapping was missing from the logs operators actually act on. Flap events, at times more than 100,000 per hour across the network, were filtered out before anyone looked. [B]

That is a reasonable choice if your goal is a quiet pager. It is the wrong choice if your goal is a reliable network. The self-clearing alert that comes back next Friday isn't noise. It is the clearest evidence you will ever get of a fault nobody has found yet.

Calling chronic outages "noise" is a recipe for SLA violations, a low-quality network and upset customers.

Noise reduction and gremlin hunting want opposite things from the same alerts, and you need both. Rapax keeps them apart. Correlation and deduplication decide what an operator sees right now. Chronic detection works on the full alert history, including every occurrence that was folded under another alert.

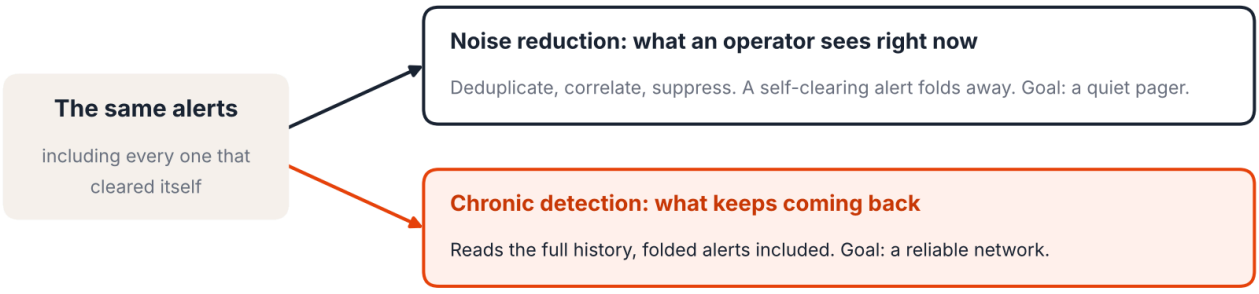


Figure 1. Two jobs for the same alerts. Correlation and deduplication decide what an operator sees now; chronic detection reads the full history.

4. What a gremlin really costs

The cost of one gremlin is spread across so many budgets that nobody sees it whole. Here are the pieces; Section 9 adds them up.

$150–1,000
per truck roll, by market and distance [C]

17–20%
of telecom dispatches estimated to find no fault [C]

60%
of U.S. consumers would look elsewhere after three outages [B]

SLA credits. Credits are typically a percentage of the affected circuit's monthly recurring charge. Contracts differ on what counts as an outage. Some set minimum outage durations. Others count every incident, and some service guides promise 100% availability, where any interruption earns a credit. Depending on the contract, a one-second drop is an outage.

Chronic-outage clauses: the lemon law for circuits. Many carrier SLAs define a "chronic" circuit and give the customer the right to terminate it. The thresholds are explicit:

- One ISP allows termination after four or more outages of at least two hours each within 30 days.
- A regional fiber provider allows it after four credit-eligible issues in 30 days.
- A Midwest carrier sets the bar at a seven-day outage, or fourteen cumulative days in twelve months.
- A Tier 1 carrier's ethernet SLAs make terminating a chronic circuit the customer's sole remedy. [B]

The industry has already defined chronic as "X outages in Y days." What most operators lack is a way to see a circuit approaching that line before their customer does.

"Prove it" clauses. Many SLAs treat a service as available if the customer won't release the circuit for testing. That puts the burden on reproducing the fault, which is exactly what a gremlin denies you. It is the contract version of the vendor telling you to prove it in the field.

Truck rolls. Industry sources put the cost of a truck roll at $150 to $1,000, depending on market and distance. They also estimate that 17–20% of telecom dispatches find no fault. [C] commonly cited industry estimates; we found no primary study. An academic study of a U.S. cable operator, covering more than 70,000 modems, estimated that 38.5% of dispatches sent the wrong type of technician. [B/C] A gremlin produces no-fault-found visits by design, because every dispatch arrives after the fault has cleared.

Parts swapped on hope. When nobody can find the fault, the equipment gets replaced. Telecom has little public data on this. Aerospace measures it carefully: 20–50% of avionics removals find no

fault, and those no-fault-found removals are estimated to cost an airline more than $180,000 per aircraft per year. [B] analogy rather than telecom data

Engineering time. Senior engineers get pulled from planned work into vendor escalations, war rooms and “stand in the room” visits. No public study measures this for intermittent network faults, but your own ticket history will show it.

Customers. Repeat outages drive churn. In a 2026 survey, 60% of U.S. consumers said they would look for an alternative provider after three outages. Broadband customers with multiple service issues were 43% likely to switch within a year, against 7% of customers without issues. [B] consumer stated intent Enterprise buyers rarely say so in a survey. They invoke the chronic clause.

The customer’s own losses. For the enterprise on the other end of the circuit, downtime costs far more than the credit. In ITIC’s 2024 survey, more than 90% of mid-size and large enterprises put the cost of one hour of downtime above $300,000. A 2024 study by Oxford Economics for Splunk estimated that downtime costs the Global 2000 $400 billion a year, about $9,000 per minute. [B] self-reported surveys That is why five-nines commitments exist, and why customers enforce them.

5. What teams try today, and why it stalls

1. **Probe everything.** Active test probes and full-coverage performance monitoring give excellent data on latency, loss and jitter, at a price. Analysts size the 2025 telecom service-assurance market anywhere from $3.9 billion to $12.3 billion; the spread itself shows how differently they define the market. [C] Probes are valuable, but they are expensive to place everywhere, and a gremlin is by definition somewhere you weren’t looking.
2. **Collect everything and hire a data scientist.** This is the big-data answer: centralize every log and counter, then bring in an outside analytics team to crunch it. It can work. It is also slow and expensive, and it usually starts after the customer has escalated, while the SLA clock keeps running. We found no public data on what these engagements cost; your procurement history will tell you.
3. **Escalate to the vendor.** The vendor asks you to reproduce the fault. A fault that clears in thirty minutes and skips some weeks can’t be reproduced on request.
4. **Swap the part.** This is cheap, fast and frequently wrong, as the no-fault-found data above shows.
5. **Put a person in the room.** This is the method that finally works, in 2014 and in Aberhosan (Section 6). It is also the most expensive way to learn a single fact: the time of day.

Most NOCs also keep a “chronic list” or watch list. It is the right idea, but it is maintained by hand, usually in a spreadsheet, and it is only as good as the memory of whoever updates it.

The common thread is that every approach either adds data you don’t have yet, or waits for a person to notice a pattern across weeks of tickets.

The pattern is usually already in your alert history.

6. A second gremlin: the 7 a.m. television

For 18 months, broadband in Aberhosan, a village in mid-Wales, failed every morning at 7 a.m. Engineers visited repeatedly and replaced cable, but the problem stayed. Finally Openreach's specialist team stood in the village before dawn with a spectrum analyser. At 7 a.m. they caught a burst of electrical interference. The source was a second-hand television that a resident switched on every morning. [B]

Openreach's engineers noted that this kind of interference is less rare than people assume. The decisive clue wasn't in the spectrum analyser, though. It was the time of day, which had been sitting in the fault reports for a year and a half. A periodicity test on those fault timestamps would have flagged a significant daily pattern months earlier, and told the team when to be there.

18 months
of broadband failures in one village [B]

7 a.m.
every morning: the clue in every fault report

1 television
switched on each morning by a resident

7. How Rapax chronic detection works

Rapax 2.0 runs a chronic-detection agent inside its analytics engine every five minutes. It runs alongside baselining (every minute), anomaly detection and forecasting, and operators can also trigger it on demand. Each cycle has five steps.



Figure 2. The chronic-detection cycle. It runs every five minutes over the last 56 days of alert history, and operators can also run it on demand.

Step 1: Find candidates. A single query over the last 56 days of alert history counts the distinct down alerts per *identity*. An identity is a device, an interface and an alert category taken together. Any identity with fewer than three is dropped.

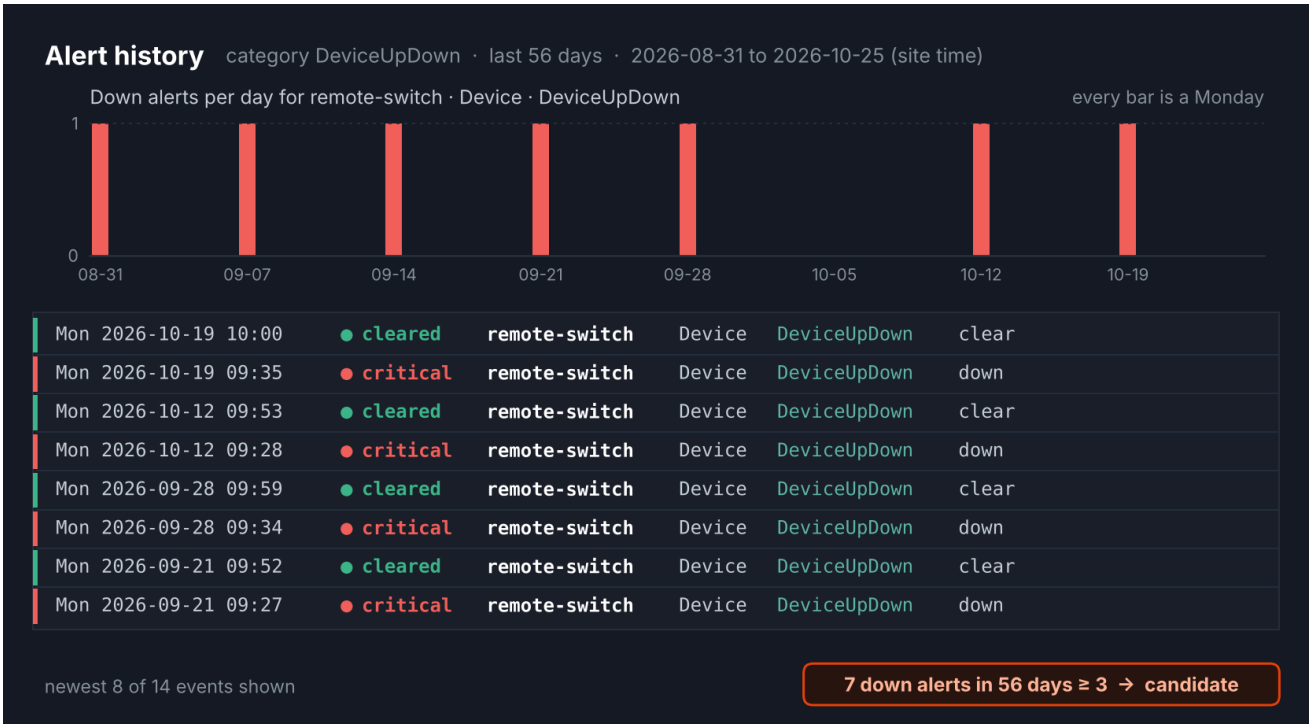


Figure 3. Step 1 on the lab system: seven down alerts for one identity in 56 days, every one on a Monday morning, so it becomes a candidate. The week of 5 October had no outage. Diagram drawn from the lab system's alert history, laid out like the Rapax Logs view.

Step 2: Turn alerts into episodes. An episode runs from an alert's first occurrence to its first clear. When the router comes back up, the alert clears, and the next unplug creates a new alert and a new episode. A repeat that never clears just raises the count and stays one episode. Every episode records its start time and its time to repair.

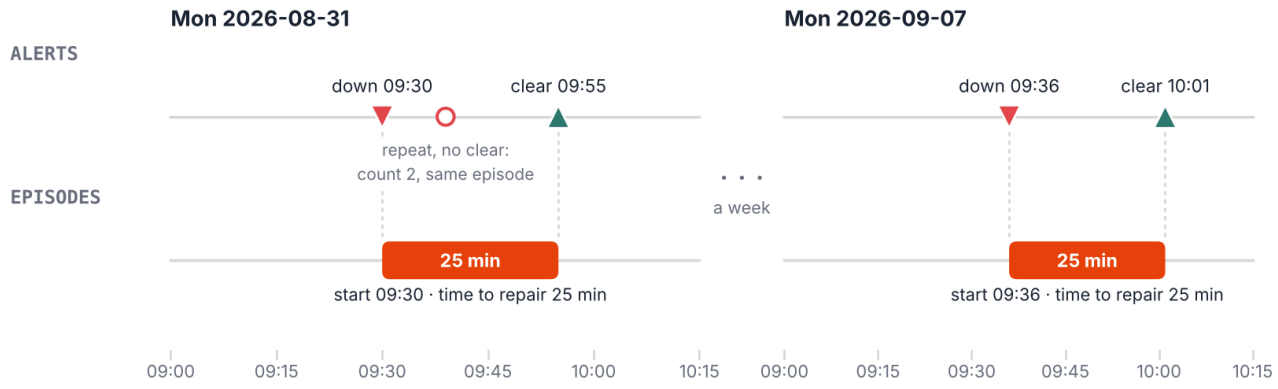
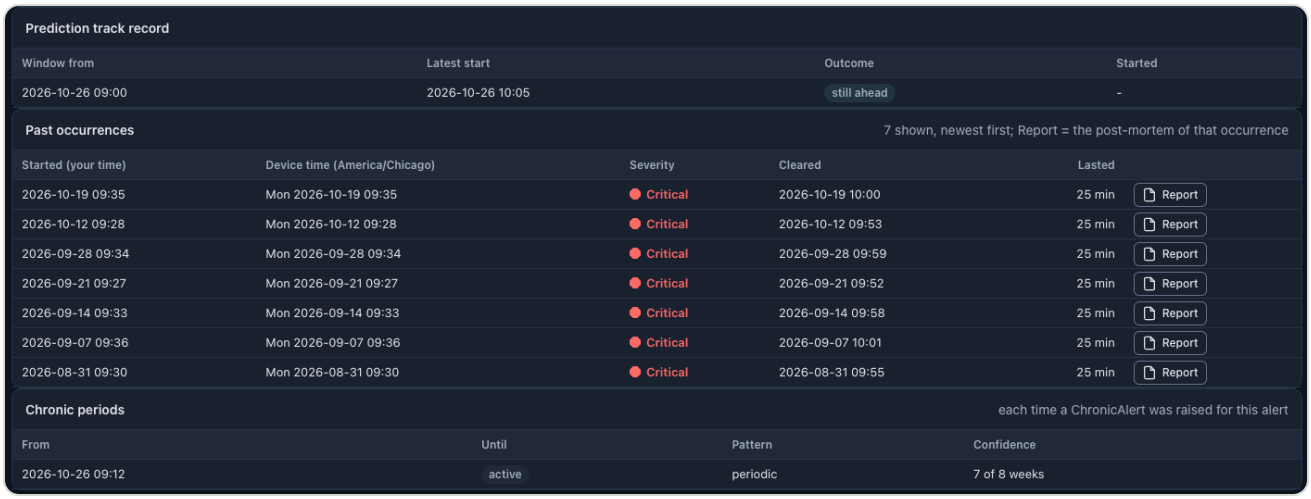


Figure 4. Step 2: alerts become episodes. An episode runs from the first down alert to the first clear; a repeat before the clear only raises the count. Times are the lab switch's first two occurrences; the repeat is added for illustration.



Prediction track record

Window from	Latest start	Outcome	Started
2026-10-26 09:00	2026-10-26 10:05	still ahead	-

Past occurrences — 7 shown, newest first; Report = the post-mortem of that occurrence

Started (your time)	Device time (America/Chicago)	Severity	Cleared	Lasted	
2026-10-19 09:35	Mon 2026-10-19 09:35	Critical	2026-10-19 10:00	25 min	Report
2026-10-12 09:28	Mon 2026-10-12 09:28	Critical	2026-10-12 09:53	25 min	Report
2026-09-28 09:34	Mon 2026-09-28 09:34	Critical	2026-09-28 09:59	25 min	Report
2026-09-21 09:27	Mon 2026-09-21 09:27	Critical	2026-09-21 09:52	25 min	Report
2026-09-14 09:33	Mon 2026-09-14 09:33	Critical	2026-09-14 09:58	25 min	Report
2026-09-07 09:36	Mon 2026-09-07 09:36	Critical	2026-09-07 10:01	25 min	Report
2026-08-31 09:30	Mon 2026-08-31 09:30	Critical	2026-08-31 09:55	25 min	Report

Chronic periods — each time a ChronicAlert was raised for this alert

From	Until	Pattern	Confidence
2026-10-26 09:12	active	periodic	7 of 8 weeks

Figure 5. The episodes as Rapax records them: every past occurrence of the lab switch with its start in your time and the device's time, when it cleared and how long it lasted, each linked to its post-mortem. The same report tracks predictions and each period the ChronicAlert was active. Lab data.

Step 3: Run three detectors. The episodes for each candidate go to three detectors in the same pass. Each looks for a different shape of trouble over a different time scale, and any one of them is enough to raise a ChronicAlert. The periodic detector does one more thing: it works out when the next episode is due.

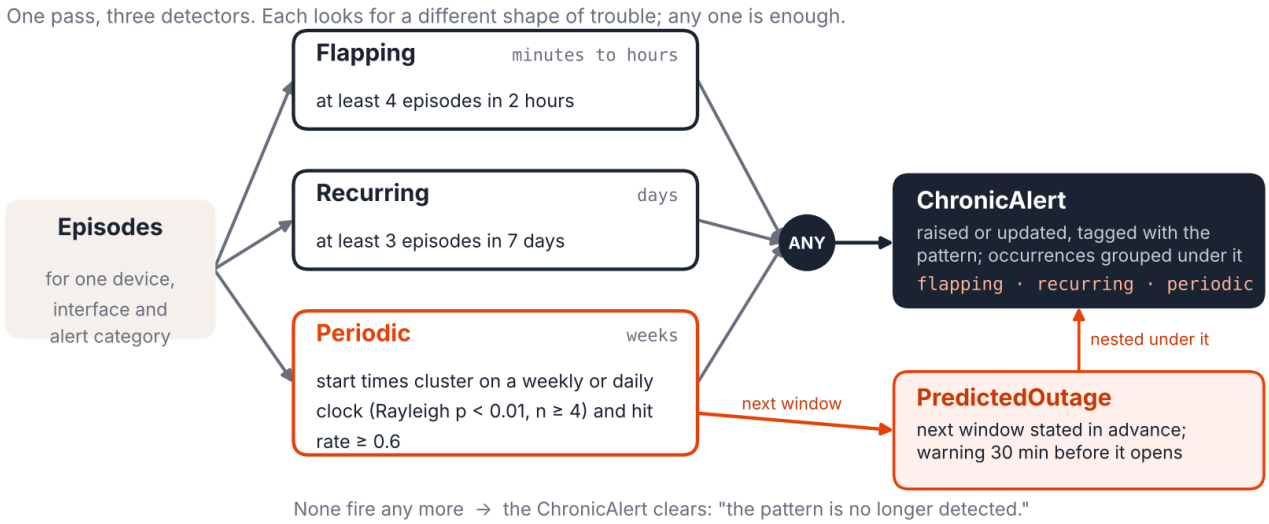


Figure 6. Step 3: three detectors in one pass. Flapping and recurring count episodes; periodic tests when they start. Any one firing raises or updates the ChronicAlert. A periodic finding also gives the next expected window and its PredictedOutage, which is nested under the ChronicAlert.

Table 2 sets out the default rules.

Table 2. The three detectors

Detector	Catches	Default rule
Flapping	Instability over minutes to hours	At least 4 episodes in 2 hours
Recurring	Chronic repeats over days	At least 3 episodes in 7 days
Periodic	Gremlins that follow the calendar	Start times clustered on a daily or weekly cycle, with at least 4 episodes

Flapping and recurring are count rules. They are cheap, immediate and easy to explain (“went down six times in two hours”). The recurring rule has the same “X in Y days” shape as the chronic clauses in your contracts.

Neither count rule would ever catch the Friday router, because once a week never reaches three in seven days or four in two hours. That is why the periodic detector exists.

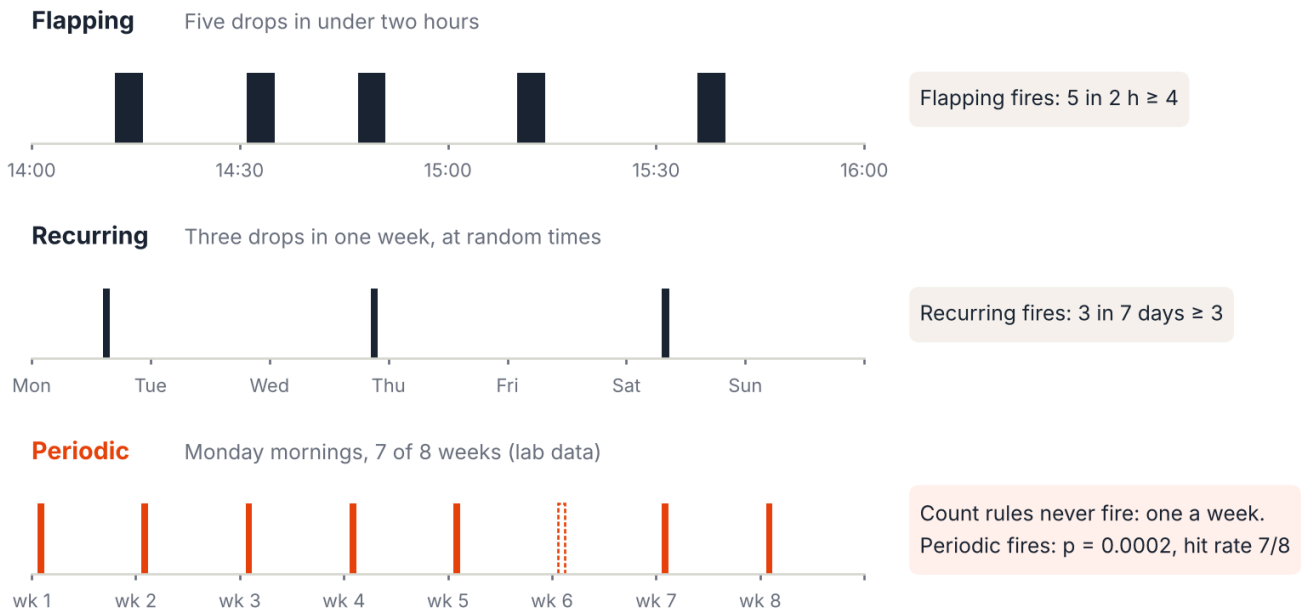


Figure 7. What each detector sees. The flapping and recurring rows are illustrations. The periodic row is the lab switch: one episode a week, so neither count rule would ever flag it.

Step 4: Decide the alerts. If any detector fires, Rapax raises or updates a ChronicAlert on the same device and interface. The ChronicAlert is tagged with the pattern (flapping, recurring or periodic) and records total down time, mean time to repair and the last occurrence.

When no detector fires any more, the ChronicAlert clears with “the pattern is no longer detected.” That also serves as evidence that a fix worked. State is kept between cycles, so a restart doesn’t duplicate or lose alerts.

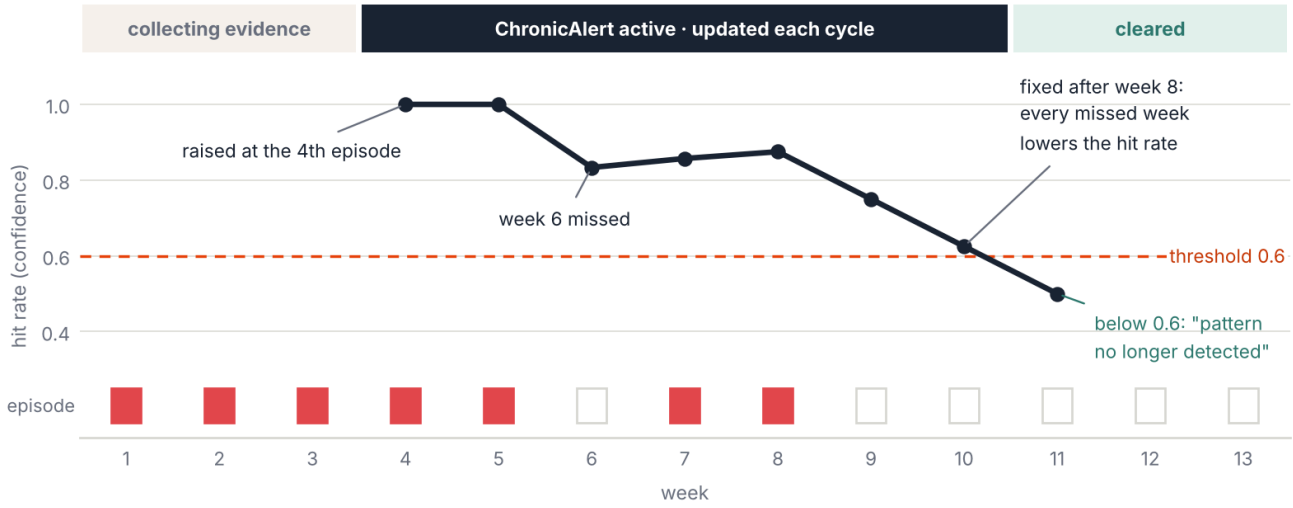


Figure 8. Step 4: the life of a periodic ChronicAlert. It is raised at the fourth weekly episode and updated each cycle. After the fault is fixed, each missed week lowers the hit rate over the 56-day window, and the alert clears when it falls below 0.6. Illustration, computed with the default rules.

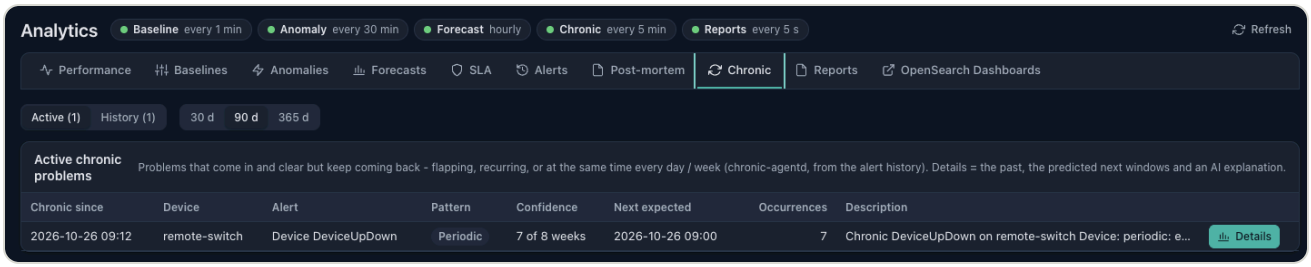


Figure 9. Active chronic problems in Rapax: the device, the alert, the pattern, the confidence and the next expected window. Lab data.

Step 5: Correlate. The correlation engine places the identity's active occurrences, and any predicted outage, under the ChronicAlert, but only if they have no other active parent. If the whole site loses power, the power alert keeps the router. A ChronicAlert never overrides a real cause.

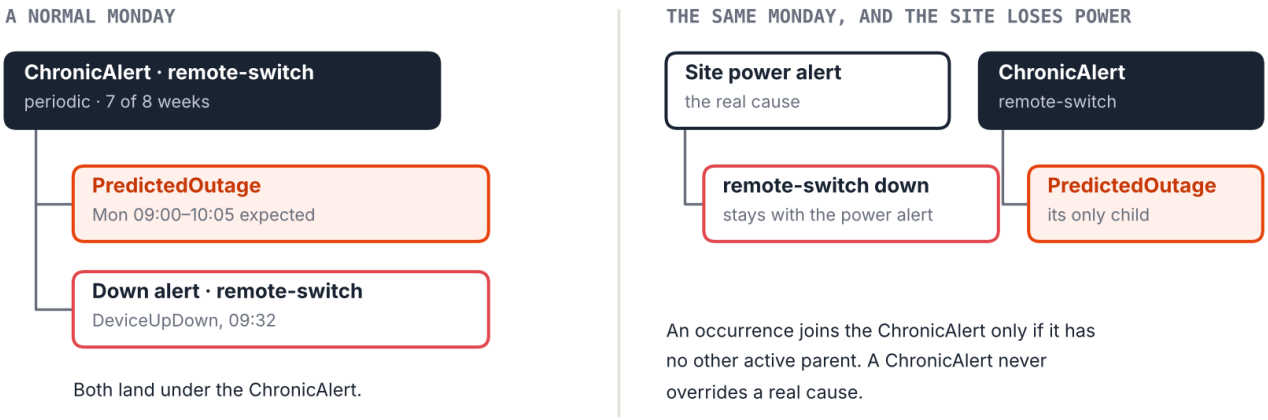


Figure 10. Step 5: correlation. On a normal Monday the down alert and the prediction land under the ChronicAlert. If the site loses power, the power alert keeps the switch's down alert, because a ChronicAlert takes only occurrences with no other active parent. Illustration.

How the periodic detector finds the Friday router

Start times become points on a clock. Each episode's start time is placed on two clocks: a weekly one, where a full turn is seven days, and a daily one, where a full turn is 24 hours. Rapax uses each site's local time. That way 3:05 p.m. stays 3:05 p.m. across a daylight-saving change, instead of jumping an hour and smearing the pattern. A network that spans time zones is judged on each site's own clock.

The Rayleigh test asks whether the points bunch together. Each start time becomes a point on the clock face, and the detector averages them as arrows of equal length. The length of that average arrow, R, is near 1 when the start times cluster and near 0 when they are spread at random. The test turns R and the number of episodes into a p-value: the probability of seeing that much clustering by chance. A pattern must reach $p < 0.01$ to count.

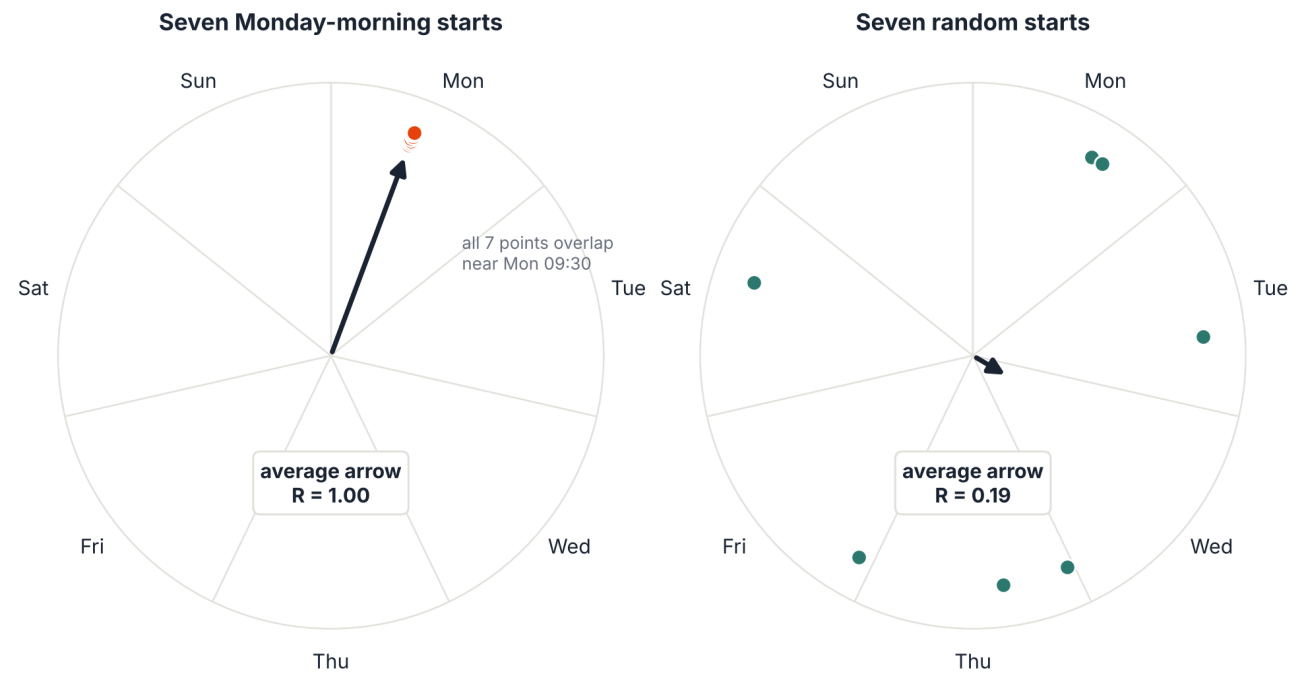


Figure 11. Illustration. Each start time is a point on the weekly clock. Seven Monday-morning starts give an average arrow of length R = 1.00; seven random starts give R = 0.19. The Rayleigh test turns R and the number of episodes into a p-value.



Figure 12. Start times by weekday and hour, in the site's local time. All seven episodes of the lab switch began on a Monday between 09:00 and 10:00. Lab data.

The hit rate decides between weekly and daily, and becomes the confidence. A Friday-afternoon pattern also clusters on the daily clock, since every start falls in the afternoon. The hit rate breaks the tie. It asks: of the cycles since the first episode, how many had an episode inside the expected window?

The window adjusts to how spread out the start times actually are. A gremlin that strikes "sometime between 2 and 5" is matched as reliably as one that strikes at 3:05 sharp.

Table 3. Illustrative example: the Friday router over eight weeks, 6 episodes

Cycle	Cycles counted	Hits	Hit rate	Result
Weekly	8 weeks	6	0.75	Passes (threshold 0.6)
Daily	about 56 days	6	about 0.11	Fails

The result reads the way an engineer would say it: *every Friday, about 2–5 p.m. local, lasting about 30 minutes; 6 of the last 8 weeks.* The Fridays when nothing happens are accounted for, because a 75% hit rate clears the 60% bar.

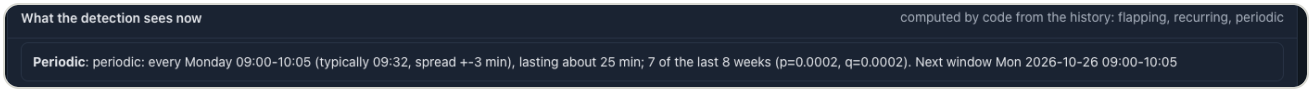


Figure 13. The same kind of reading from a lab system: a switch that went down most Monday mornings, 7 of the last 8 weeks (p = 0.0002). Lab data.

Prediction. Once the pattern is confirmed, the ChronicAlert and the chronic report state the next expected window, days before it arrives. Thirty minutes before the window opens, Rapax raises a PredictedOutage warning showing the expected time and the confidence ("6 of 8 weeks"). If the router drops, the down alert and the prediction both land under the ChronicAlert. The prediction clears once the window has passed.

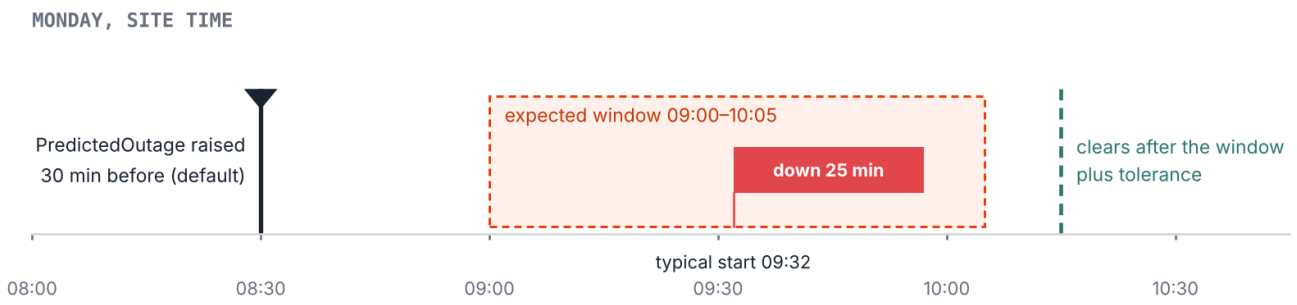


Figure 14. The prediction timeline, using the lab pattern and the default 30-minute lead time. The warning clears after the window plus a tolerance; where that line falls here is illustrative.

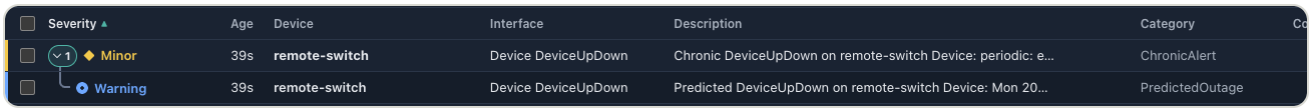


Figure 15. The ChronicAlert in the console, with its PredictedOutage nested under it. In this lab run the pattern was first detected at 09:12, after the predicted window had opened, so the warning was raised then, ahead of the expected 09:32 drop. Lab data.

When it stops. If the cleaners start using another outlet, each missed Friday lowers the hit rate. Once it falls below 60%, the ChronicAlert clears.

Detection takes evidence. The periodic detector needs at least four episodes, judged only on the cycles since the first one. For a weekly gremlin, that means roughly four to six weeks of history. The requirement is deliberate: it is the minimum evidence needed to avoid crying wolf.

That evidence is gathered automatically from the first occurrence, so nobody has to remember last Friday. If your existing alert history is loaded through a custom import during onboarding, a gremlin that has been running for months can be visible on day one. Without an import, the evidence builds from the day Rapax starts collecting.

Planned work is not excluded yet. Rapax 2.0 doesn't know your maintenance windows. A weekly change window that takes a device down will be reported as a periodic pattern, because it is one. Check a periodic finding against your change calendar before you dispatch anyone.

8. Prepared, not surprised

When a gremlin follows the clock, Rapax tells you when to be there. When it doesn't, Rapax tells you where to look and gives the technician the whole history.

For calendar-locked gremlins, dispatch shifts from chasing to scheduling. Dispatch knows the expected window days ahead and gets a warning before it opens. Someone can be on site when it happens: a technician, a camera or a site contact. One well-timed visit replaces months of trucks that arrive after the fault has cleared.

Some gremlins don't follow a clock: the flapping link, or the circuit that drops three times a week at random times. The flapping and recurring detectors still find them, and Rapax puts what a technician needs in one place:

- **The ChronicAlert** groups every occurrence of the identity under one parent, with total down time, MTTR and last occurrence.
- **The post-mortem view** for any parent alert shows the correlation tree, the services affected and how long each was down. It also shows a single timeline of alert changes, service state, incidents and operator actions.
- **An AI-drafted summary** of the event appears alongside the facts. It is explicitly marked as a draft to check against them.
- **The SLA view** tracks every service against its target over 24 hours, 7, 30 and 365 days. It shows outages, mean time to recover and the remaining error budget, so a gremlin eating into a five-nines budget is visible in seconds rather than at month-end.

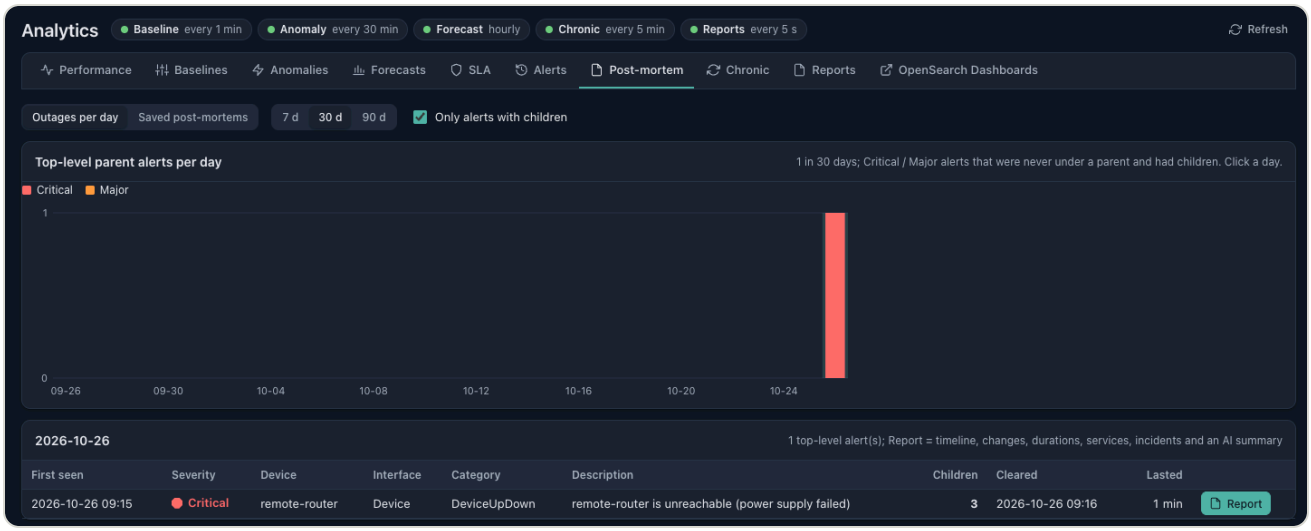


Figure 16. The Post-mortem tab: top-level outages per day, each with a report covering the timeline, changes, durations, services, incidents and an AI summary. Lab data.

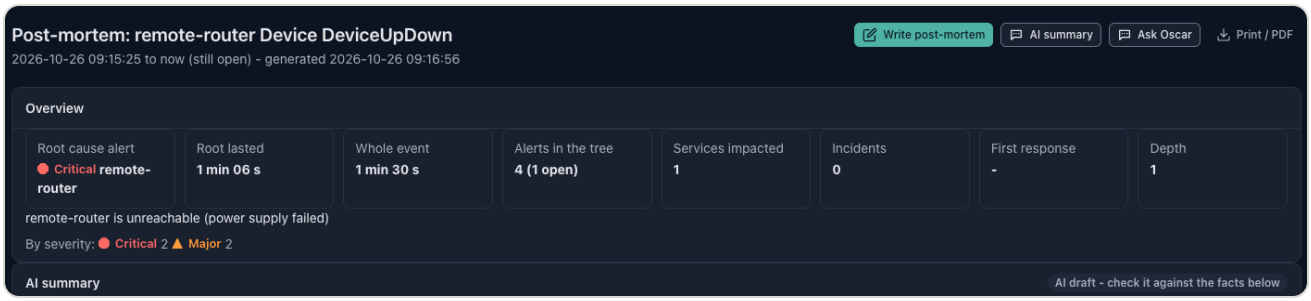


Figure 17. A post-mortem for a parent alert: root cause, how long it lasted, the alerts in its tree, services impacted and incidents. The AI summary is marked as a draft to check against the facts. Lab data.

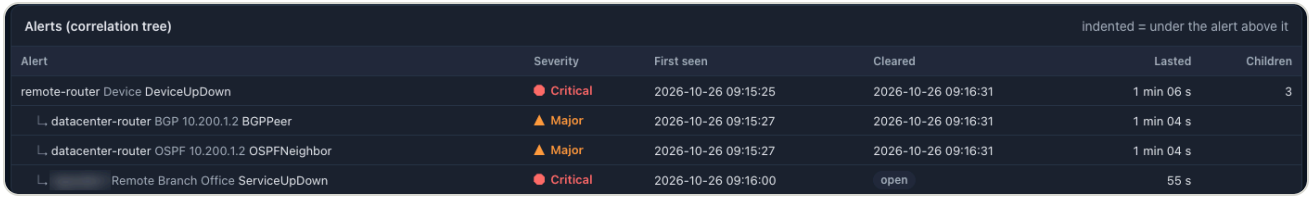
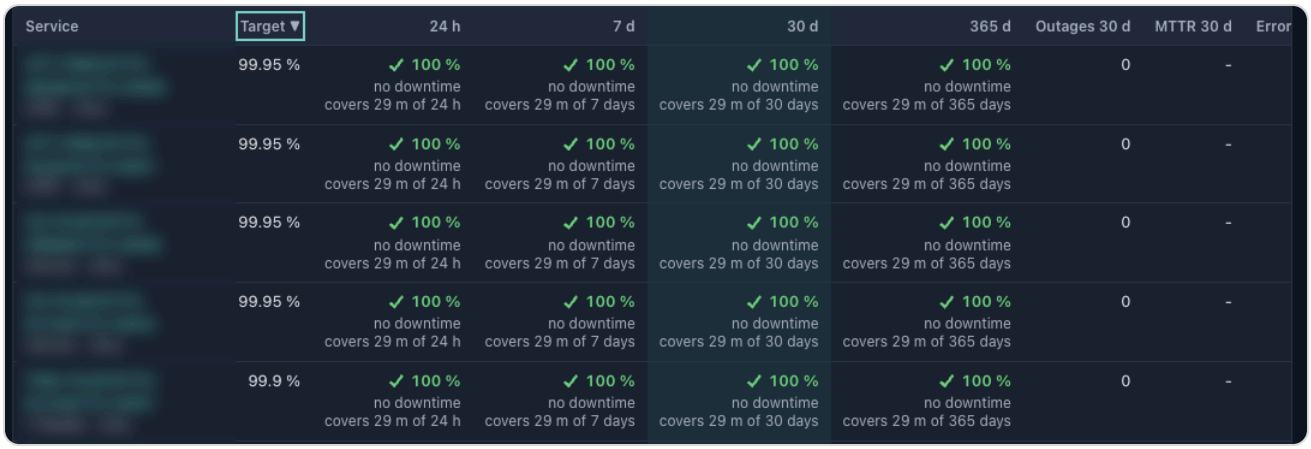


Figure 18. The correlation tree in the same post-mortem: the router failure with the BGP and OSPF peer alerts and the service outage under it. Lab data; an internal host name is blurred.



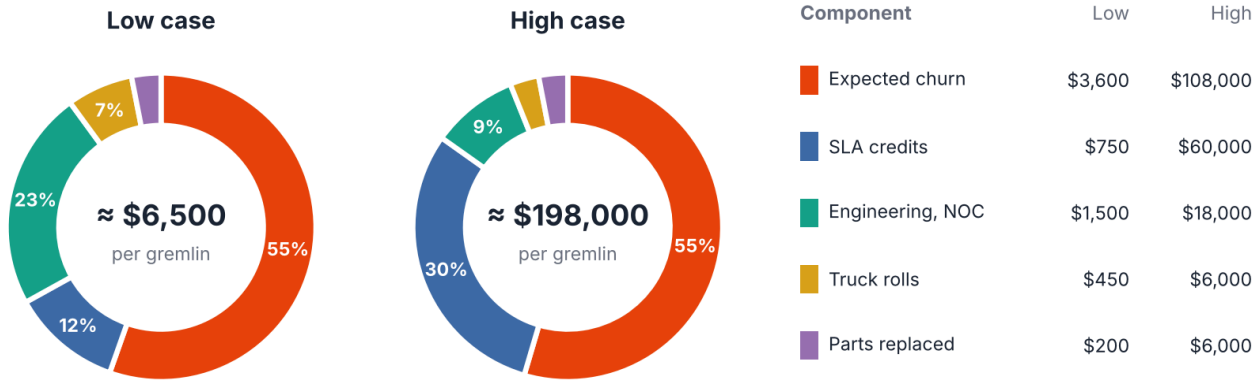
Service	Target ▼	24 h	7 d	30 d	365 d	Outages 30 d	MTTR 30 d	Error
[redacted]	99.95 %	✓ 100 % no downtime covers 29 m of 24 h	✓ 100 % no downtime covers 29 m of 7 days	✓ 100 % no downtime covers 29 m of 30 days	✓ 100 % no downtime covers 29 m of 365 days	0	-	
[redacted]	99.95 %	✓ 100 % no downtime covers 29 m of 24 h	✓ 100 % no downtime covers 29 m of 7 days	✓ 100 % no downtime covers 29 m of 30 days	✓ 100 % no downtime covers 29 m of 365 days	0	-	
[redacted]	99.95 %	✓ 100 % no downtime covers 29 m of 24 h	✓ 100 % no downtime covers 29 m of 7 days	✓ 100 % no downtime covers 29 m of 30 days	✓ 100 % no downtime covers 29 m of 365 days	0	-	
[redacted]	99.95 %	✓ 100 % no downtime covers 29 m of 24 h	✓ 100 % no downtime covers 29 m of 7 days	✓ 100 % no downtime covers 29 m of 30 days	✓ 100 % no downtime covers 29 m of 365 days	0	-	
[redacted]	99.9 %	✓ 100 % no downtime covers 29 m of 24 h	✓ 100 % no downtime covers 29 m of 7 days	✓ 100 % no downtime covers 29 m of 30 days	✓ 100 % no downtime covers 29 m of 365 days	0	-	

Figure 19. The SLA view: each service against its target over 24 hours, 7, 30 and 365 days. Service names blurred. Lab data.

Rapax doesn't claim to name the root cause; a person still finds the cleaning crew or the television. What changes is that they start with the evidence, in the right place, at the right time.

9. The honest numbers

Every gremlin is different, so we don't quote a single savings figure. Instead, here is a transparent model of what one gremlin costs an operator. It uses the sourced ranges above, plus assumptions you can replace with your own.



Component	Low	High
Expected churn	$3,600	$108,000
SLA credits	$750	$60,000
Engineering, NOC	$1,500	$18,000
Truck rolls	$450	$6,000
Parts replaced	$200	$6,000

Modeled [C]; assumptions in Table 4.

Figure 20. Where the money goes in the modeled cost of one gremlin. Expected churn is the largest share in both cases; SLA credits are second in the high case. Modeled [C]; the assumptions are in Table 4.

Table 4. Modeled cost of one gremlin, operator view [C, modeled]

Cost component	Low case	High case	Basis
SLA credits	$750 ($1,000 monthly charge × 25% × 3 months)	$60,000 ($5,000 monthly charge × 100% × 12 months)	Monthly charge and credit levels assumed
Truck rolls	$450 (3 × $150)	$6,000 (6 × $1,000)	Industry range [C]
Engineering and NOC time	$1,500 (20 h × $75)	$18,000 (120 h × $150)	Loaded rates assumed
Parts replaced	$200 (1 × $200)	$6,000 (3 × $2,000)	Assumed
Direct costs	**$2,900**	**$90,000**	
Expected churn loss	$3,600 (10% × $36k lifetime value)	$108,000 (60% × $180k lifetime value)	Churn rates bracketed by survey data; lifetime value = monthly charge × 36 months (assumed)
Total per gremlin	**≈ $6,500**	**≈ $198,000**	

How to read the model:

- **Churn dominates the high case.** If your contracts make termination hard, discount that line. If they include chronic clauses, it is the line that matters most.
- **Credits can fall to zero** where a contract sets minimum outage durations. That moves the cost into churn; it doesn't remove it.
- **The customer's own losses aren't included.** At ITIC's $300,000-plus per hour, one to five hours of cumulative gremlin downtime costs the customer $300,000 to $1.5 million. That exposure is why customers enforce SLAs.

Fleet illustration [C]. Suppose my field estimate holds and 2% of a 5,000-circuit enterprise base carries a chronic fault. That is 100 gremlins, costing $0.65 million to $19.8 million a year at the per-gremlin range above. Every input is an assumption except the sourced ranges.

The useful question isn't which end of that range you're on. It is how many chronic identities are in your alert history right now, and the evaluation answers that.

Where Rapax changes these numbers. Finding a gremlin sooner shrinks every line: fewer blind dispatches, fewer parts swapped, fewer months of credits and less exposure to a chronic clause. We haven't yet published production results for chronic detection, so we don't claim a percentage reduction here. We would rather show you your own data.

About the numbers

Every figure in this paper is graded:

- [B] marks a benchmark from a named external source, listed under Sources.
- [C] marks an estimate or model, with its assumptions shown.

Survey data is self-reported and, in several cases, vendor-sponsored. Aerospace figures are analogies, not telecom data. Where we couldn't find a source, we say so. This edition contains no Rapax production figures for chronic detection.

10. Objections we hear

"Our AIOps platform already does this." Correlation and noise reduction are mature, and Rapax does them too. The better question is whether your platform keeps the self-clearing alerts it suppresses, tests them for daily and weekly clustering, and tells you when the next window is. Noise reduction and chronic detection work on the same alerts toward opposite goals.

"Periodic gremlins are rare." We don't know how common they are in your network, and nobody has published a good number. What is documented is that network failures follow strong time patterns. A study of the seven largest U.S. broadband ISPs found that 29–75% of disrupted address blocks were disrupted *only* during the weekday overnight maintenance window. [B] Human schedules, maintenance, polling and backup jobs all run on clocks. Periodic is also only one of three detectors; flapping and recurring don't need a clock at all.

"Probes give better data." For latency, loss and jitter, they do. Chronic detection runs on the alert history you already have, and its job is to tell you where to put a scarce probe, analyser or technician, and when. The two are complementary, not competing.

"This is just threshold tuning." Thresholds and flap dampening act on single events over seconds or minutes. Chronic detection looks across weeks, the same horizon as your repeat-trouble metrics and chronic-circuit clauses. Tuning reduces noise; it doesn't find causes.

"Correlation isn't causation." Agreed. Rapax narrows the search and times the visit, then hands the evidence to the person who will find the cause. It doesn't claim to name the root cause.

"A statistical test on a handful of points will cry wolf." Four episodes is the minimum by design, and at four points the Rayleigh test is very strict: the start times must be tightly clustered to pass (Appendix A). A pattern must also clear the hit-rate test. The one systematic false lead is your own planned work: Rapax 2.0 doesn't exclude maintenance windows, so check periodic findings against your change calendar.

Sidebar: NetOps legends

Every NOC has a legend. A favorite from the broadband era is an aggregation router that would reboot if you SNMP-walked it three times in fifteen minutes. We couldn't verify that story, so we present it as legend.

The behavior behind it is documented, though. Cisco security advisories describe:

- devices that reload in response to specific SNMP polls;
- a switch family that reloaded under a heavy stream of SNMP traffic, a flaw found while resolving a customer support case;
- a firewall platform whose SNMP process leaked memory under heavy polling until the device restarted. [B]

Your own monitoring and maintenance schedule can be the gremlin. When restarts line up with a nightly backup or a weekly job, the periodic detector sees the pattern. When they arrive in bursts, the flapping detector does.

11. Getting started

Chronic detection ships in Rapax 2.0 and runs on the alert history Rapax already collects.

1. **Bring your history, if you want results sooner.** Alert history from your current tools can be loaded through a custom import during onboarding, so the detectors have eight weeks of data from the start. Without it, detection begins as history accumulates; a weekly pattern needs about four to six weeks.
2. **Review the ChronicAlerts.** Each one names the device, interface, pattern and confidence. For periodic patterns, it also gives the next expected window.
3. **Tune the defaults, not the method.** Count windows, minimum episodes, the hit-rate bar and lead time live in one configuration file. Check periodic findings against your change calendar; Rapax 2.0 doesn't exclude maintenance windows automatically.
4. **Schedule the visit.** For your first periodic gremlin, have someone on site during the predicted window.
5. **Close the loop.** When the fix lands, the pattern stops and the ChronicAlert clears.

The free evaluation

Send us eight weeks of alert history. We'll run chronic detection on it and show you the gremlins already in your network, before your customers find them.

Request a free evaluation: rapax.app/evaluate
Or book 15 minutes: cal.com/shawn-ennis

Appendix A: The math

Phase. For an episode starting at site-local time t, the phase on a cycle of length P (24 h or 168 h) is $\theta = 2\pi \cdot (t \bmod P) / P$.

Mean resultant length. For n episodes, $R = \left| (1/n) \Sigma e^{i\theta} \right|$, where $0 \le R \le 1$.

Significance (Rayleigh test). Rapax uses Zar's small-sample approximation, with $R_n = nR$: $p \approx \exp(\sqrt{(1 + 4n + 4(n^2 - R_n^2))} - (1 + 2n))$. For large n, this approaches $p \approx e^{(-nR^2)}$. A periodic candidate requires $p < 0.01$ and $n \ge 4$. Figure 11 shows R for clustered and random start times.

Why four episodes is strict. At $n = 4$, the smallest possible p-value, reached when all four start times coincide, is about 0.0076. Only very tightly clustered start times pass.

Consider four Friday starts spread across 2.5 hours. On the weekly clock that is about 5.4°, giving $R \approx 0.999$ and $p \approx 0.008$, so the pattern passes. The same starts on the daily clock span 37.5°, giving $p \approx 0.012$, so that cycle fails at four episodes. With six episodes, the weekly pattern reaches $p \approx 0.0004$.

Hit rate. This is the fraction of cycles since the first episode that contain an episode inside the expected window around the mean phase. The window adapts to the observed spread. The pattern must reach a hit rate of at least 0.6, and the hit rate is reported as the confidence.

Prediction. The next window is the next local occurrence of the mean phase. Its expected duration is the median repair time of past episodes. A PredictedOutage warning is raised 30 minutes before the window by default, and clears after the window plus tolerance.

What Rapax 2.0 does not do. We would rather you hear the limits from us:

- **It detects daily and weekly cycles only.** Monthly and seasonal patterns need longer history and aren't detected today.
- **It works best with a single peak.** The Rayleigh test is most powerful when a pattern has one peak per cycle. A gremlin that strikes at both 7 a.m. and 7 p.m. weakens it.

- **It counts down alerts.** Performance degradations that never produce a down alert aren't counted as episodes.
- **It needs history.** The default window is eight weeks, so alert retention must cover at least that.
- **It doesn't know your maintenance windows.** Planned work that recurs on a schedule is detected as periodic. Check findings against your change calendar.
- **It doesn't import history on its own.** Loading alerts from other tools is a custom import, set up per source during onboarding.

Sources

1. Turner, D., Levchenko, K., Snoeren, A., Savage, S. "California Fault Lines: Understanding the Causes and Impact of Network Failures." ACM SIGCOMM 2010. Includes the citation of Markopoulou et al. on the Sprint backbone. https://cseweb.ucsd.edu/~klevchen/tlss-sigcomm10.pdf
2. Gill, P., Jain, N., Nagappan, N. "Understanding Network Failures in Data Centers." ACM SIGCOMM 2011 (Microsoft Research). https://www.microsoft.com/en-us/research/wp-content/uploads/2017/01/sigcomm11netwiser.pdf
3. FCC Common Carrier Bureau, Quality of Service Monitoring Report, 1998 (repeat trouble definitions). https://transition.fcc.gov/Bureaus/Common_Carrier/Reports/FCC-State_Link/Monitor/mr98-9.pdf
4. Public Utility Commission of Texas, service quality filing (repeat trouble report rates). https://interchange.puc.texas.gov/Documents/46729_189_968962.PDF
5. Incident-response platform product page, event intelligence / transient alert auto-pause (vendor claim; not named per editorial policy).
6. iTel Networks, Service Level Agreement. https://itel.com/sla/
7. Pathwayz Communications, Connectivity Service Level Agreement. https://pathwayz.com/connectivity-service-level-agreement/
8. 123.Net, Service Level Agreement. https://www.123.net/service-level-agreement/
9. CenturyLink (Lumen), Ethernet Private Line SLA. https://www.centurylink.com/legal/docs/Custom_EPL_SLA_111511_v1.pdf
10. Firstsource, "Truck Roll Reduction" (industry glossary). https://www.firstsource.com/insights/glossary/truck-roll-reduction
11. Asentria, "Reducing Telecom Truck Roll Operation Costs with Automation." https://www.asentria.com/reduce-telecom-truck-roll-operation-costs-using-automation/
12. Hu, J., Zhou, Z., Yang, X. "TelApart" (cable broadband fault diagnosis), arXiv:2412.09740, 2024. https://arxiv.org/pdf/2412.09740
13. Raza, A., Ulanskyi, V. *Applied Sciences* 11(2):715, 2021 (no-fault-found rates in avionics maintenance). https://www.mdpi.com/2076-3417/11/2/715
14. Copernicus Technology, "Introducing No Fault Found." https://www.copernicustechnology.com/index.php/technical-services/introducing-no-fault-found
15. Xurrent / Dynata consumer outage survey, 2026. https://www.prnewswire.com/news-releases/new-data-three-outages-and-customers-will-look-elsewhere-302802475.html
16. Parks Associates / TechSee, via Telecompetitor, 2025. https://www.telecompetitor.com/in-home-wi-fi-problems-create-customer-churn-risk-parks-associates/
17. ITIC, 2024 Hourly Cost of Downtime Survey. https://www.calyptix.com/wp-content/uploads/ITIC-2024-Hourly-Cost-of-Downtime-Survey-Results-Part-2.pdf
18. Splunk and Oxford Economics, "The Hidden Costs of Downtime," 2024. https://www.splunk.com/en_us/newsroom/press-releases/2024/conf24-splunk-report-shows-downtime-costs-global-2000-companies-400-billion-annually.html
19. Telecom service assurance market estimates, 2025 (IMARC, GMI, Fortune Business Insights, Mordor Intelligence). https://www.imarcgroup.com/telecom-service-assurance-market

20. Openreach, “Second-hand TV wipes out broadband for entire village,” 2020. https://www.openreach.com/news/second-hand-tv-wipes-out-broadband-for-entire-village/
21. APNIC Blog, “When the Internet goes down: improving the detection of disruptions,” 2018. https://blog.apnic.net/2018/11/26/when-the-internet-goes-down-improving-the-detection-of-disruptions/
22. Cisco Security Advisories: cisco-sa-ftd-snmp-dos-R8ENPbOs (CVE-2020-3533); cisco-sa-20180307-550x (CVE-2018-0209); cisco-sa-20180620-n4k-snmp-dos (CVE-2018-0299). https://sec.cloudapps.cisco.com/security/center/
23. Landler, L., Ruxton, G., Malkemper, E. Comparison of the Rayleigh and Hermans–Rasson tests for circular uniformity, 2019 (PMC). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6686250/

© 2023–2026 Rapax, a Citus Technologies, LLC company. www.rapax.app