by tufin

AKiPS™

NETWORK MONITORING GUIDE

The Network *Visibility Gap*

WHY RESOLUTION AND RETENTION DEFINE YOUR INCIDENT RESPONSE

When something goes wrong on the network, the first question is always the same: what happened, and when exactly did it start? For most network teams, the honest answer is constrained by their monitoring data. If monitoring data is averaged, low-resolution, or retained at full granularity for only 30 days, the window for accurate reconstruction closes fast. High resolution, un-averaged data sharpens incident response, enables meaningful trend analysis, and provides the data quality any AI-driven monitoring layer requires.

INSIDE THIS GUIDE

Network teams are working with lower resolution data than every other team that depends on them

For years, five-minute or ten-minute polling was standard. Storage was expensive. Pollers weren't built to handle the load. The storage constraint is gone, but many monitoring platforms never caught up. Most still collect interface data at five minute intervals and average it before storage. The resulting graphs look smooth and clean. But averaging erases the spikes, drops, and brief anomalies that actually occurred. The graph ends up displaying utilization values the network never actually hit.

SUB-MINUTE VISIBILITY CHANGES WHAT YOU CAN PROVE DURING AN INCIDENT

When an incident happens, the first question is always when exactly it started. With ten minute polling, the answer is a range: "somewhere between 2:50 and 3:00." With one minute polling, the answer is a timestamp. When you can tell the application team that something changed at 3:01pm rather than somewhere around 3, the log review window drops from thousands of entries to dozens. A 60-second bracket is the difference between a 20-minute resolution and a 4-hour war room.

Some failure modes are invisible at low polling rates:

Micro-bursts	An interface that spikes to 100% utilization for 10 seconds will look like 80% on a 5-minute poll. One-minute polling shows it for what it is.
Packet drops	Even when a micro-burst doesn't show on a utilization graph, the resulting error counts and discards will. They appear in the one-minute window where they occurred.
Intermittent errors	A cable starting to fail, an SFP degrading. These show up as sporadic error spikes. At a 5-minute average, they disappear entirely.

WHY AVERAGED DATA FAILS AT THE MOMENT YOU NEED IT MOST

Many platforms average polled data before storing it. Averaging reduces storage costs. The tradeoff is that full-resolution data is never stored, so there is no precise historical record to go back to. Go back six weeks on an averaged platform and you're looking at hourly averages, sometimes daily averages. The incident your security team needs to reconstruct, where something changed on a specific port at a specific time, is gone.

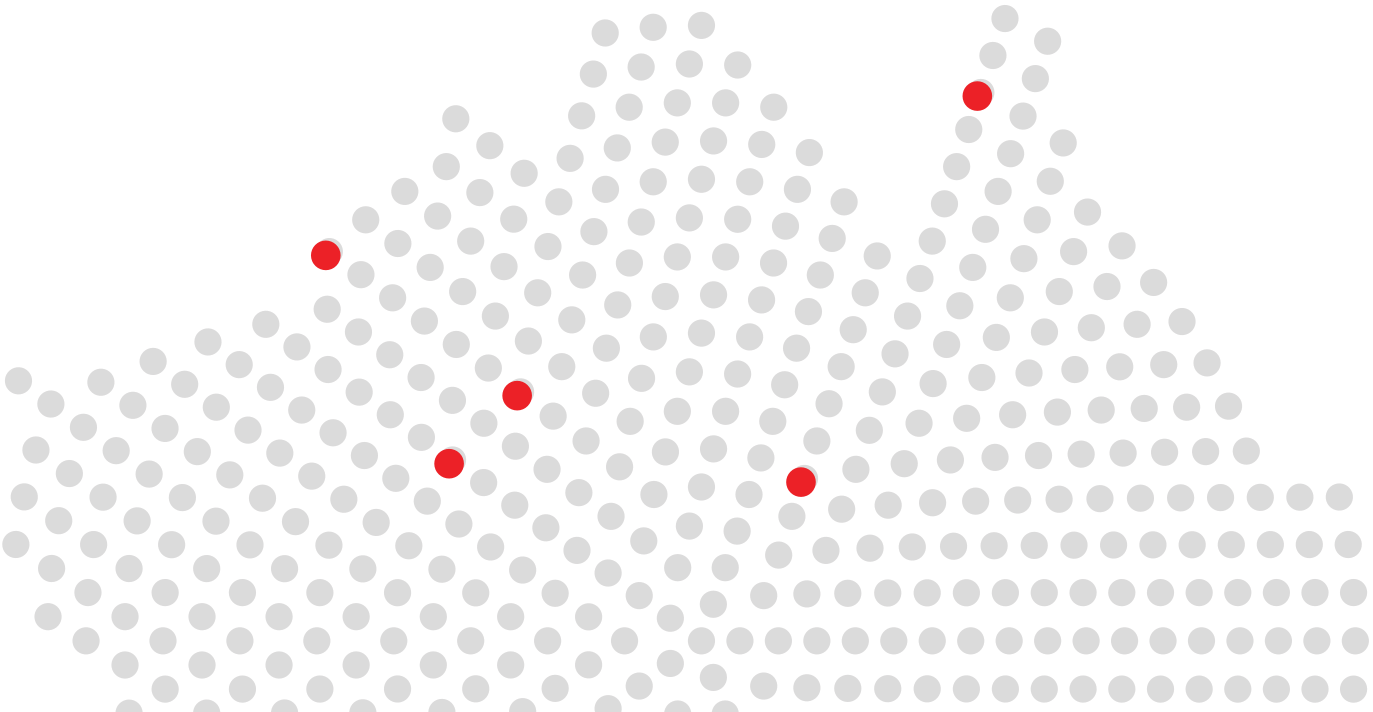
Raw, un-averaged data stored at full resolution means you can go back three years and see exactly what changed, on which interface, at which minute. Compliance investigations need

that precision to establish what happened and when. Capacity planning needs months of real traffic data, not smoothed trend lines, to forecast accurately. And any AI or machine learning layer needs the same thing: enough granular history to distinguish normal behavior from anomalies.

	AVERAGED DATA	RAW DATA
Graphs	Smooth	Accurate
Storage cost	Lower	Higher, but manageable
Incident timestamps	"Around 3pm"	"3:01pm"
Data after 30 days	Daily averages	One-minute resolution for 3 years
Micro-bursts	Invisible	Visible
Trend lines	Smooth	Actual network behavior

THE STORAGE ARGUMENT IS A RED HERRING

The concern that high-frequency polling will overwhelm storage or add significant network overhead is real for platforms not built for high-frequency polling. AKIPS was designed from the ground up for efficient high-frequency polling. Polling overhead is minimal. Storage is manageable. Platforms that struggle with high-frequency polling weren't built to handle it at scale.



Poor data quality creates alert fatigue, and alert fatigue creates blind spots

Static threshold alerting has a simple logic: if a metric crosses a line, send an alert. For binary conditions, that logic works. For most of what network engineers actually deal with, it generates more noise than signal. When alert volume is high enough and the signal-to-noise ratio poor enough, engineers stop responding. The system has trained them that most alerts do not require action.

THE ALERT FATIGUE REALITY

Network engineers routinely receive dozens or hundreds of monitoring alerts per day. Many teams only respond to alerts that persist or repeat. Some have stopped reading certain alert streams entirely. Ignoring alert streams is a rational response to a broken alerting system.

WHY STATIC THRESHOLDS MISS THE FAILURE MODES THAT MATTER MOST

Static thresholds work when the problem is simple and binary: a circuit is up or it's down; CPU is above 95% for five minutes. For anything more nuanced, static thresholds create more noise than signal.

Interface errors	Some errors on any interface are normal. A threshold that fires on any errors creates noise. A threshold set too high misses the early signs of failure.
Memory utilization	A switch running at high memory is not necessarily a problem. A switch where memory has been climbing steadily for weeks or months is a problem. That pattern is often a vendor-acknowledged memory leak requiring a reboot.
Utilization patterns	A link at 90% during business hours is expected. A link that hits 90% at 3am on a Tuesday is worth investigating. Static thresholds treat both identically.

WHAT TREND-DRIVEN MONITORING CATCHES THAT THRESHOLDS CANNOT

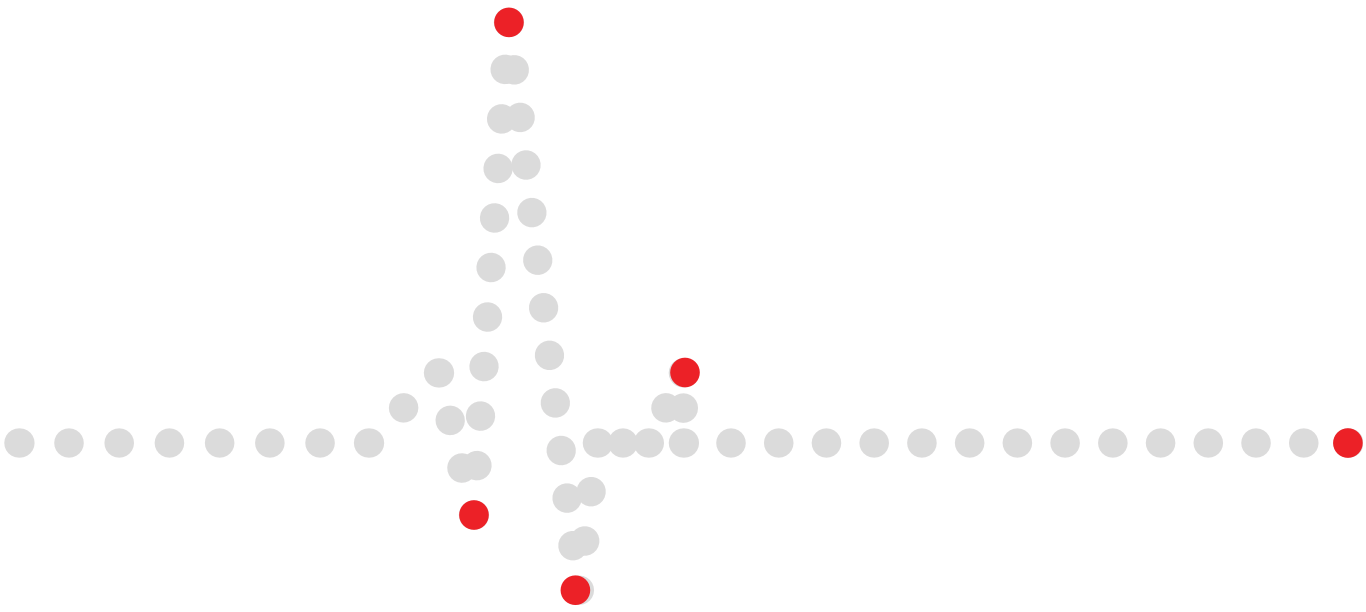
Trend-driven monitoring adds a layer on top of threshold alerting that catches slow degradation, gradual increases, and behavioral changes over time. In practice, that means an alert that reads "interface X has had increasing error counts for two weeks and the rate is accelerating" rather than "interface X has errors." Accelerating error counts indicate a failing SFP or damaged cable, caught before it causes an outage.

COMMON FAILURE MODES THAT TREND MONITORING CATCHES

Memory leaks on network devices (slow climb over weeks or months). SFP degradation (gradually increasing error counts). Capacity creep (utilization that slowly approaches a ceiling). BGP instability (increasing flap frequency over time). Interface errors that increase after a physical change.

THE DATA QUALITY PREREQUISITE FOR TREND ANALYSIS AND AI-DRIVEN MONITORING

Trend-driven monitoring, and any AI or machine learning layer built on top of monitoring data, depends entirely on the quality of the underlying data. You can't detect a trend in a memory metric if you're only looking at hourly averages. You can't identify a gradual error count increase if the data is smoothed before storage. And you definitely can't train a model on daily averages and expect it to surface meaningful anomalies. High-resolution, raw, historical data is the prerequisite for trend analysis, anomaly detection, and any AI driven monitoring layer.



The polling frequency and retention period you choose today determine what your data can do for years

Both capabilities covered in this guide, high-resolution polling and trend-driven alerting, depend on data that is accurate, granular, and stored at full resolution over time. A platform that averages data, retains full-resolution data for only 30 days, or degrades to hourly averages after 90 days limits every analytical capability you add on top, permanently.

THE THREE DATA QUALITY DECISIONS THAT DETERMINE WHAT YOUR DATA CAN SUPPORT

1

GRANULARITY: HOW OFTEN DATA IS COLLECTED

The right polling frequency differs by metric. Interface metrics need one-minute polling. Device availability pings need 15-second intervals. At one minute, a 10-second utilization spike appears in the data. At five minutes, it's averaged into the surrounding values and disappears entirely.

2

RETENTION: HOW LONG FULL-RESOLUTION DATA IS KEPT

A platform that stores one-minute data for 30 days and rolls it up to hourly has a 30 day window for anything requiring precise timestamps. A platform that stores one minute data for three years has a three-year investigative window. Network incidents are often investigated long after they occur, sometimes six weeks later. Both trend analysis and AI-driven monitoring require enough historical depth to distinguish normal behavior from anomalies.

3

STORAGE FORMAT: RAW OR AVERAGED

Averaging is a compression technique. It makes data easier to store and graph. It also removes the peaks, the anomalies, and the brief events that tend to be the most important things in the dataset. Raw data, stored exactly as it was collected, is the foundation on which meaningful trend analysis, anomaly detection, and AI-driven monitoring can be built.

WHAT TO ASK BEFORE EVALUATING ANY PLATFORM

Ask: what is the maximum polling frequency? What is the retention period at that resolution? Does the platform average data before storage, and if so, when? These three questions will tell you more about long-term value than any feature checklist.

Most platforms will fail at least one of these questions, and that failure has a direct cost

Parts 1, 2, and 3 describe what good monitoring data looks like and what it makes possible. The questions below are how you find out whether your current platform or one you're evaluating actually delivers it. Every question maps to a specific failure mode. A weak answer means that failure mode is present in your environment right now.

POLLING FREQUENCY AND RESOLUTION: WHAT YOUR PLATFORM ACTUALLY CAPTURES

What is the minimum polling interval for interface metrics, and for device availability pings?

One minute for interfaces and 15 seconds for pings are the benchmarks worth measuring against. Anything slower means micro-bursts, intermittent errors, and brief packet drop events may not appear in your data at all. Get a specific number, not a range.

Does the platform store raw polled values, or does it average data before storage?

Many vendors describe their graphs without mentioning the averaging step. If the response redirects to graph quality or visual smoothness, the platform is averaging data before storage. Push until you get a direct answer.

What is the retention period at full resolution, and when does that data get rolled up?

Retention at full resolution is not the same as total data retention. A platform that keeps one-minute data for 30 days and then rolls up to hourly has a 30-day investigative window, regardless of how many years of averaged data it stores. Vendors often quote total retention to obscure the full-resolution figure. Get both numbers and compare them.

Has increasing the polling frequency ever caused platform instability?

Platform instability under high-frequency polling is a symptom of how the platform was built, not how large the network is. A platform designed for one-minute polling handles the load without issue. A platform that wasn't will show poller lag, UI slowdowns, and missed polls under load. Ask for specific examples. The answer tells you more about long-term fit than any feature checklist.

ALERT QUALITY: WHETHER YOUR PLATFORM SIGNALS PROBLEMS OR TRAINS YOU TO IGNORE IT

Can the platform alert on rate-of-change over a configurable time window, not just when a threshold is crossed?

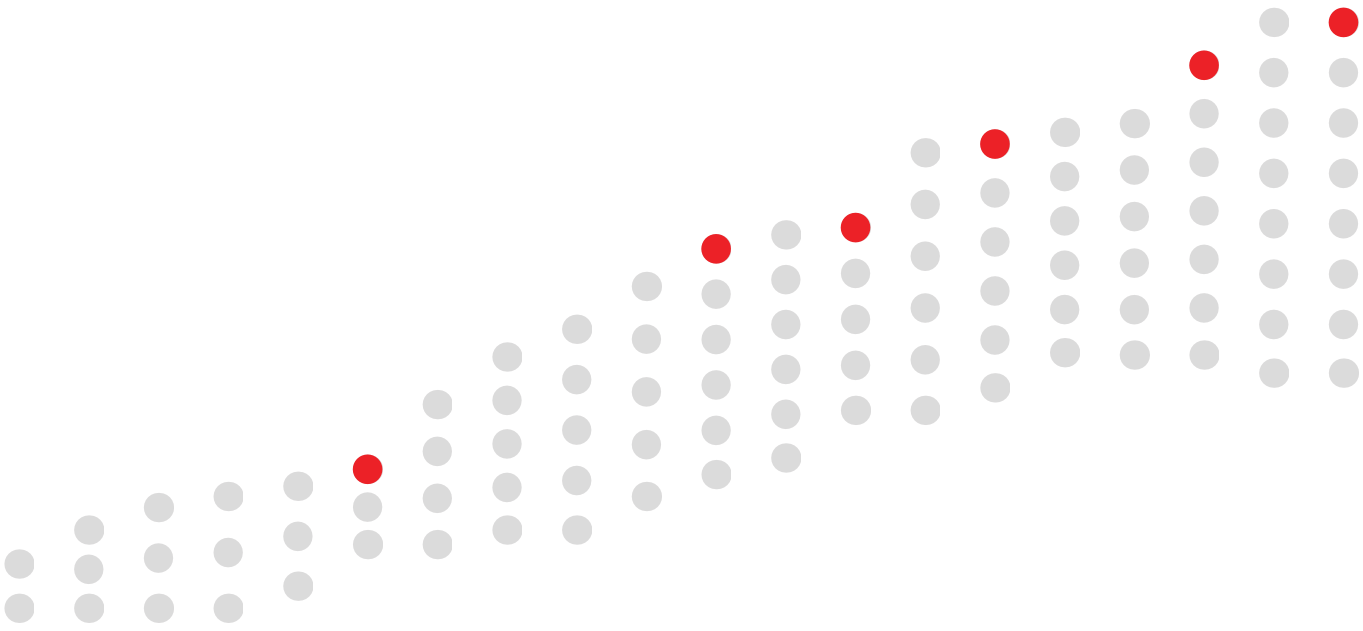
Threshold alerting fires when a metric crosses a line. Trend-based alerting fires when a metric has been moving in a direction for long enough to indicate a real problem. A platform that can only do the former will surface "interface X has errors" but miss "interface X error counts have increased 40% over the past two weeks." Confirm that trend alerting includes rate-of-change over a configurable window, not just a rolling average threshold.

What is the typical daily alert volume for a network of our size?

High alert volume is a product decision, not an inevitability. A vendor who can't give a concrete number, or whose reference number runs into the hundreds per day, is describing a system that will train engineers to stop paying attention. Ask for a real example from a comparable customer, not a capability description.

How far back can you query historical data for trend analysis?

Trend analysis is only as deep as the full-resolution data underneath it. If one-minute data is retained for only 30 days, any trend query beyond that window runs on averaged data, smoothing out the gradual patterns trend monitoring exists to catch. The historical query window and the full-resolution retention period should be the same number.



DATA ACCESS AND RESILIENCE: WHETHER YOUR PLATFORM WORKS FOR YOU OR LOCKS YOU IN

What data is available through the API, and does that include raw polled values?

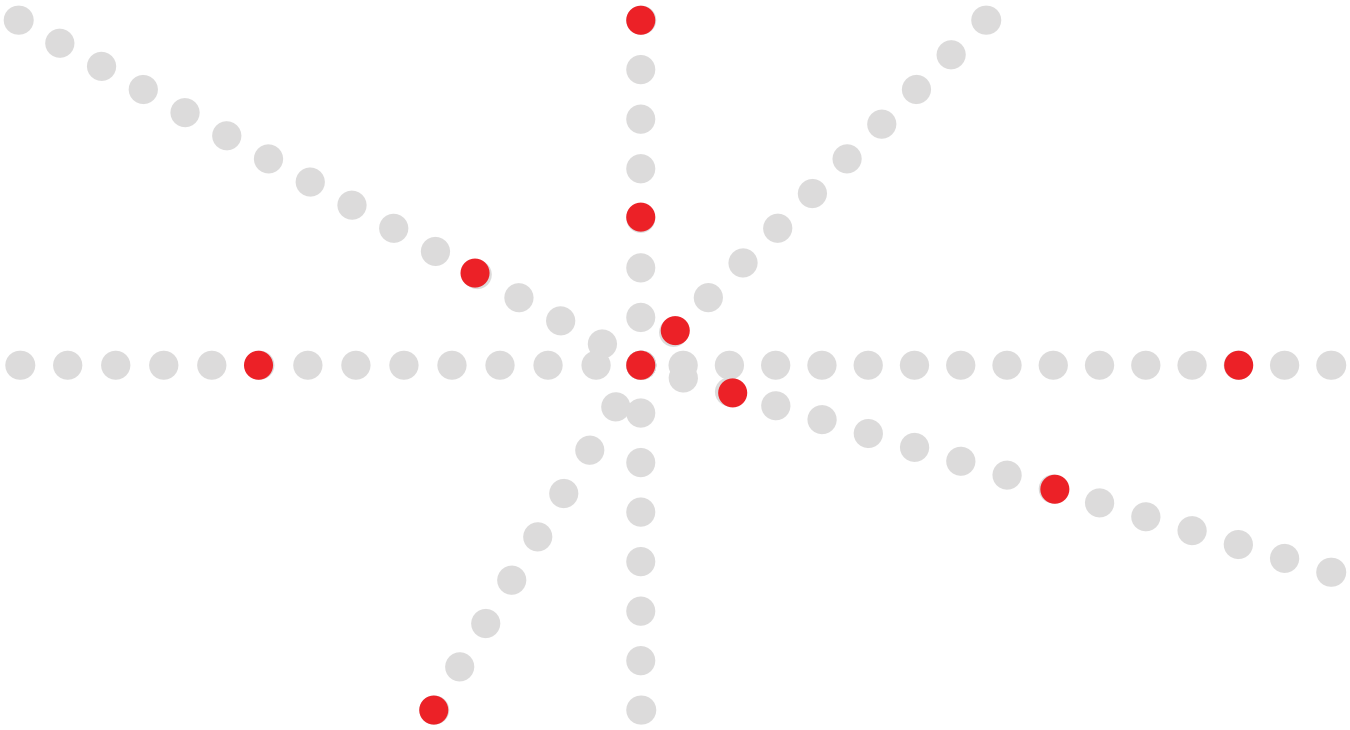
Raw historical data locked behind the UI creates a hard ceiling on what you can build on top of the platform. External tooling, custom dashboards, and AI-driven analysis layers all require API access to underlying data. A platform that only exposes aggregated exports through its API limits every integration you add on top.

Can alert data be sent natively to Slack, Teams, or PagerDuty?

Routing alerts through a separate integration layer adds failure points between the monitoring system and the engineers who need to act. Native delivery is a basic operational expectation, not a premium feature.

What happens to the platform if internet access is lost, and can it run entirely on-premises?

A monitoring platform that depends on internet connectivity goes dark during the exact conditions it exists to help you manage. For teams in regulated environments or with data residency requirements, on-premises deployment is a non-negotiable. For everyone else, the answer determines whether the platform is actually available when it matters most.



ABOUT

AKIPS

AKIPS is a network monitoring platform designed from the ground up for high frequency polling: one-minute interface metrics and 15-second device availability pings, stored as raw, un-averaged data for up to three years. AKIPS built its poller and time-series database in-house, specifically to handle large-scale networks at high polling frequencies without performance degradation.

- One-minute polling for all interface metrics, out of the box
- Raw, un-averaged data stored for up to three years
- Full API access to raw historical data
- 15-second ping resolution for device availability
- Vendor-agnostic SNMP support across thousands of device types
- Deployable entirely on-premises with no cloud dependency

See what your network actually looks like

Stand up AKIPS on your own network in under 30 minutes. One-minute polling, raw un averaged data, three years of retention, all out of the box.

AKIPS.COM/TRIAL